Project Report
VLG-2

Very Large Graphs for Information Extraction (VLG)

Detection and Inference in the Presence of Uncertainty

B.A. Miller
L.D. Weiner
A.I. Reuther
N. Arcolano
M.S. Beard
M. Wolf

21 September 2015

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Intelligence Advanced Research Projects Activity (IARPA) under
Air Force Contract FA8721-05-C-0002.

This report is based on studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This work was sponsored by the Intelligence Advanced Research Projects Activity (IARPA) under Air Force Contract FA8721-05-C-0002.

Massachusetts Institute of Technology
Lincoln Laboratory

Very Large Graphs for Information Extraction (VLG)

Detection and Inference in the Presence of Uncertainty

B.A. Miller
Group 58

L.D. Weiner
Group 45

A.I. Reuther
Group 53

N. Arcolano
M.S. Beard
M. Wolf
formerly Group 53

Project Report VLG-2

21 September 2015

Lexington Massachusetts

This page intentionally left blank.

EXECUTIVE SUMMARY

In numerous application domains relevant to the Department of Defense (DoD) and the Intelligence Community (IC), data of interest take the form of entities and the relationships between them, and these data are commonly represented as graphs. In its role as a DoD-sponsored federally funded research and development center (FFRDC), MIT Lincoln Laboratory (MIT LL) assisted the Intelligence Advanced Research Projects Activity (IARPA) with independent scientific research and analysis of uncued detection techniques for anomalous characteristics within massive graphs in which structure and content change over time, and observations may be uncertain or corrupted.

As part of the Very Large Graphs for Information Extraction (VLG) technical effort, MIT LL performed a second one-year proof-of-concept study on the impact that various uncertainty mechanisms have on detection performance within Lincoln Laboratory's Signal Processing for Graphs (SPG) framework. Several models for data corruption and obfuscation are proposed, including models from the open literature and several inferred from experience with real graph data. Mechanisms include loss of information, edges observed in error, and confusion of vertices with those having similar metadata. The quantitative impact of each uncertainty mechanism on detection performance is demonstrated in simulation, with analytical results provided for simpler models. In addition, a data framework was developed to evaluate the quantitative differences and similarities between network datasets, to ensure proper data diversity in future experiments considering multiple target datasets. Under this study, several application datasets available in the public domain were characterized in this framework. Finally, it is demonstrated that the key kernel in the SPG framework can be applied to a four-billion-vertex graph and complete in under five minutes when run on a large supercomputing cluster.

The outcomes of the experiments and data characterizations lead to the recommendation that future work use models for simulation that incorporate community structure into the background, and to emphasize techniques that enable a simple scheme for multi-observation fusion, allowing for different data with different uncertainty to be combined in a way that enables an increase in performance without making significant assumptions about the underlying model. To ensure that algorithms developed in future efforts can handle diverse data, it is recommended that both a fast-moving transaction dataset and a slower social dataset be used in future experiments, with different characteristics in their distributions of clustering coefficient, PageRank and eigencentrality, as these are the most prominent differences among the several real datasets analyzed.

This page intentionally left blank.

ACKNOWLEDGMENTS

This work is sponsored by the Intelligence Advanced Research Projects Activity (IARPA) under Air Force Contract FA8721-05-C-0002. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government.

This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

This page intentionally left blank.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION

Many applications of interest involve relationships, connections, or transactions between a large set of entities. In particular, detection of interesting sets of such pairwise interactions—those that warrant deep investigation—within a large volume of data is a key problem in many Lincoln Laboratory mission areas. For example, it would be desirable to detect computer network traffic that is consistent with botnet activity, or people in a social network planning a nefarious activity. In these applications and many others, the relational data of interest are manifest in a graph.

Graphs are a common mathematical representation for relational data, and have recently become extremely popular for encoding relationships in the vast array of data available via web crawls and online social networks. Working with graph data of these sizes, however, leaves many canonical algorithms impractical for use, especially in data with short time constraints.

With this in mind, MIT Lincoln Laboratory (MIT LL) developed a new technical area called Signal Processing for Graphs (SPG), and a spectral framework for uncued detection of anomalous subgraphs [1–4]. Building on this framework, MIT LL carried out a one-year proof-of-concept study to determine the capabilities and challenges in the detection of anomalies in extremely large graphs [5]. Under this effort, two real datasets were considered, and algorithms for data modeling and anomaly detection were developed based on the phenomena observed in those data. This study demonstrated the ability to discover botnet traffic within a set of web proxy data when state-of-the-practice techniques left this activity undetected, and showed the capability of performing a key algorithm on a billion-vertex graph using a commodity computing cluster.

In a second one-year study, MIT LL considered the impact of data uncertainty and corruption on detection performance using this framework. The notion of how to properly sample a graph has been considered in the recent past [6], but the recent increase in research on social networks is beginning to drive new work on the impact of noise, uncertainty, and corruption in graphs in the social sciences [7] as well as the mathematics, statistics, and computer science communities [8–12]. This is an area that is starting to gain traction among network science researchers, and it is important to understand the implications of noise and uncertainty on the SPG subgraph detection framework. This report documents the models for data uncertainty used, and their impact on detection performance. Models from the open literature are considered, as well as models intuited from experience with real datasets. In addition, a data characterization framework was developed to determine features of datasets that will allow verification that datasets used in practice cover the relevant graph characteristic space.

The remainder of this report is organized as follows: Section 2 provides an overview of the present study. Section 3 documents the models used for the underlying graph data and the uncertainty mechanisms. The impact of these mechanisms on detection performance is shown in Section 4. Analysis of real data is provided in Section 5, and the data characterization framework is presented in Section 6. Section 7 provides a summary and outlines future work.

This page intentionally left blank.

2. STUDY OVERVIEW

The present study is a follow-on of the first Very Large Graphs for Information Extraction (VLG) study [5], which focused on scaling algorithms developed under the SPG effort to large, dynamic graphs derived from real datasets. The objective of the new one-year study is to quantify the change in graph data due to various uncertainty mechanisms, such as missing data and noisy data. Understanding the effects of such mechanisms on detection performance is required in a well-defined experimental framework for the detection of anomalies in very large graphs. This study is intended to inform future development of large-scale systems for subgraph detection, taking into account uncertainty in the data.

Several models for data uncertainty are considered in the experiments documented here. These include missing edges, missing vertices, random insertion and removal of edges, a propagation-based sampling approach, and an approach based on similarity of vertex metadata. These models are described in Section 3.

In simulation, detection performance was evaluated with these uncertainty mechanisms in place. Each mechanism has a distinct impact on detection performance, and it is demonstrated that fusion of multiple corrupted observations can achieve the same performance as when no uncertainty mechanism is used. Performance in simulated weighted graphs is also explored, demonstrating that thresholding based on weights may severely reduce detection performance. These results are detailed in Section 4.

In the same real datasets as the prior study, mechanisms for error in the derived graphs were inferred, informing or confirming some of the proposed models (in particular random missing edges and the mechanism based on vertex similarity). Also, simulated uncertainty mechanisms were applied to the Web of Science data, demonstrating phenomena similar to what is seen in very simple models for random graphs. These results are presented in Section 5.

A data framework was also developed in order to understand the varying dimensions of characteristics of very large graphs. A baseline set of characteristics was proposed, and several real datasets from publicly available research repositories were characterized with respect to this set, demonstrating the features that vary most greatly from network to network. Simulated graphs were also generated, which match the real graphs in many of the chosen characteristics. Building on the prior effort, additional parallel processing software was developed, enabling fast computation of the principal eigenvector of the residuals matrix of a four-billion-vertex graph on a supercomputing cluster in under five minutes. The results of this analysis are shown in Section 6.

This page intentionally left blank.

3. MODEL DEVELOPMENT

The purpose of this task was to develop statistical models for graph uncertainty that can be used in simulation to demonstrate the effect of uncertainty on detection performance in the SPG framework. In this section, the kinds of uncertainty of interest in this context are outlined, and several models are proposed.

There are a few dimensions to uncertainty in this context, as outlined in Table 1. In any scenario in which graph data is of interest, there is a true (or "latent") network representing the relationships interesting to the analyst. However, this graph may not be observable, and the observed graph may not match exactly with the relationships of interest. For example, if the relationship of interest is "are friends," this would be impossible to verify for an extremely large network. Using a proxy for friendship, such as being connected on an online social network like Facebook, makes the collection tractable, but not entirely accurate (not everyone is a member of Facebook, some members may connect with casual acquaintances as well as friends, etc.). This sort of uncertainty is referred to as "observation mismatch." In addition, random errors may occur in the collection process, which is called "collection uncertainty." Either kind of uncertainty may impact the observed vertices or edges.

3.1 LATENT GRAPH MODELS

Several models have been used under the VLG and SPG efforts as data generators. Some key generative models are:

- The *Chung–Lu* model gives each vertex a weight, and the probability of an edge between to vertices is the product of their weights. This model is used to create graphs with a given degree distribution, but cannot create community structure.

Uncertainty Type	Description	Example
Latent Graph	Graph encoding true relationships of interest	Friendship network
Observation Mismatch	Difference between observable and desired relationship	Observable proxy to friendship, e.g., Facebook
Collection Uncertainty	Errors due to imperfections in the collection process	Sensor noise Entity extraction errors
Vertex Uncertainty	Uncertainty regarding an entity's true identity and attributes	Ambiguous names
Edge Uncertainty	Uncertainty regarding correctness of observed connections	Edge connected to the wrong person/document

TABLE 1

Dimensions of uncertainty in graphs.

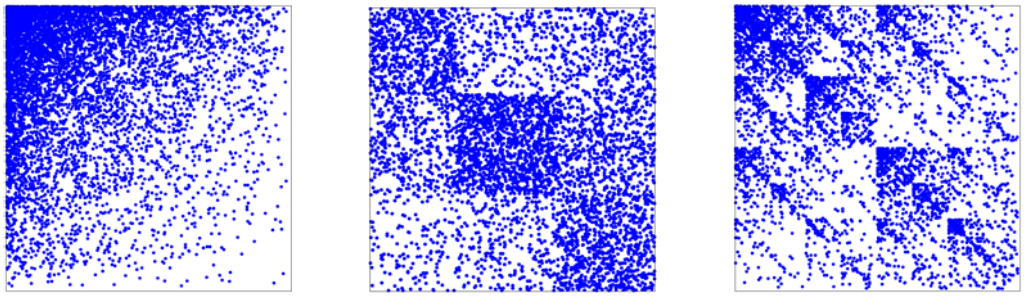

Figure 1. Adjacency matrices for candidate latent graph models: Chung–Lu (left), Stochastic Blockmodel (center), and R-MAT (right).

- The *Stochastic Blockmodel* creates a graph with community structure, where each vertex is assigned to a community and edge probabilities vary between pairs of communities. This model can create strong community structure, but does not create skewed degree distributions, which are often seen in real graphs.
- The *Stochastic Kronecker Graph* family of graph models, such as the Recursive Matrix (R-MAT) model, defines edge probabilities in a recursive manner by splitting the graph in half, defining edge probabilities within and between each half, and recursing within each of those possible subsets. This creates graphs with a fractal-like structure, and mild community structure in addition to skewed degree distributions.

Sparsity patterns of the adjacency matrices of these graphs are shown in Figure 1.

In the previous VLG study, a model was suggested for incorporating vertex metadata into edge probabilities. This was a generalized linear model (GLM) that modeled edge probabilities in a logistic regression framework, in which the probability of an edge from vertex i to vertex j is

$$p_{ij} = \frac{1}{1 + \exp\left(-x_{ij}^T \beta\right)},$$

where x_{ij} is a vector of attributes associated with vertex pair (i, j), and β are weights. Fitting this model to real data, however, is too computationally complex for extremely large graphs. Putting some restrictions on the types of attributes (specifically, allowing vertex attributes and categorical vertex-pair attributes), and assuming that the graph is sparse, this model can be modified such that analysis is tractable for very large graphs [13]. One possible parameterization of this model allows for community structure and skewed degree distributions, thus enabling a generative model with characteristics seen in real networks. Detection performance in using this model as a dynamic background is shown in receiver operating characteristic (ROC) curves in Figure 2, which also demonstrates the impact of model mismatch.

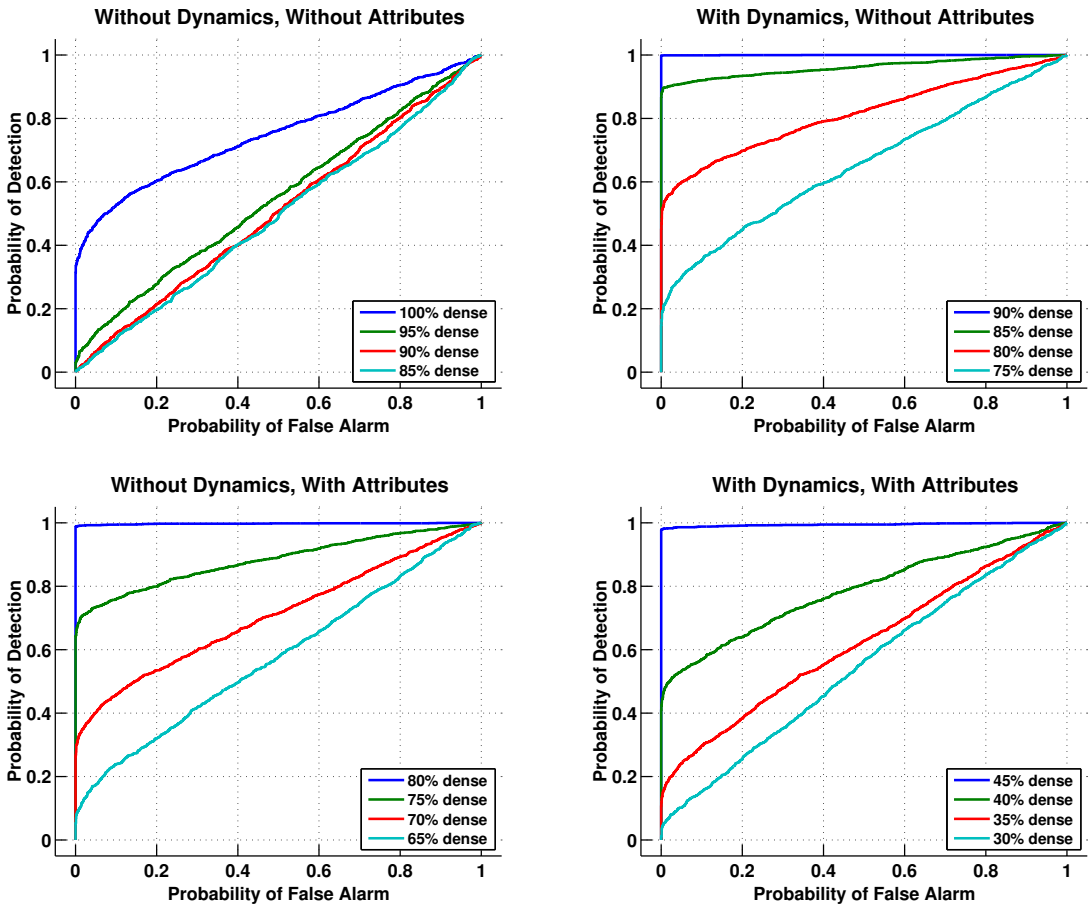

Figure 2. Detection performance in a dynamic attributed graph. A much weaker subgraph is detectable when considering both dynamics and community structure than when considering either property alone.

3.2 UNCERTAINTY AND CORRUPTION MODELS

The latent graph is observed through some imperfect mechanism, which is modeled as the addition and removal of vertices and edges. The uncertainty mechanism models used in the experiments in this report are the following:

Uniform Edge Removal This mechanism is relatively simple: each edge that exists in the latent graph is observed with a fixed probability p, and no false edges are observed. While simple, this model has the advantage of enabling some theoretical analysis (see Section 4.1) and being consistent with one application-specific form of data loss (see Section 5.2). This model is equivalent to performing an entrywise logical AND of the adjacency matrices for the latent graph and an Erdős–Rényi random graph with probability parameter p.

Uniform Edge Error This model considers every pair of vertices in the graph, and creates an "edge error," i.e., a missing edge or a false edge. This is equivalent to performing an entrywise logical XOR of the adjacency matrix for the latent graph and an Erdős–Rényi random graph with probability parameter p.

Degree-Biased Edge Error This model is a modification of the previous algorithm that normalized the number of edges per vertex based on the vertex degree. That is, the expected number of edge errors associated with a given vertex is proportional to the number of edges adjacent to that vertex. This model is equivalent to performing an entrywise logical XOR of the adjacency matrix for the latent graph and a Chung–Lu random graph where the weight of each vertex is proportional to its degree.

Observed Subgraph In some cases, some of the vertices in a graph will not be observable. This mechanism allows observation of the induced subgraph of the latent graph of a randomly selected subset of vertices (i.e., the selected vertices and all connections between them). This is similar to *egocentric sampling* [7].

Snowball Sampling A network may be estimated by sampling a few vertices, then recursing on their neighbors. This simulates sampling by, for example, forwarding surveys [7]. Each vertex is randomly assigned a probability of "forwarding" the sampling algorithm, and, upon being chosen, selects a subset of its neighbors to also be included in the sample.

Similarity-Based Uncertainty An error mechanism seen in real data (see Section 5.1) is confusion between vertices with similar metadata. This mechanism is modeled by a stochastic matrix S that is based on a similarity measure between all pairs of vertices. The ijth entry in the matrix, s_{ij} represents the probability that vertices i and j will be confused for one another when the graph is observed (or, for $i = j$, the probability that the correct vertex will be recorded). In this model, the expected value for the observed graph, given the true graph with adjacency matrix A, is $S^T AS$.

Any of these mechanisms may model either observation mismatch or collection error, and the choice of the proper uncertainty mechanism will likely be application specific.

3.3 CHALLENGES AND RECOMMENDATIONS FOR FUTURE WORK

The primary lesson learned from the model development task is to use a model with community structure, as most real networks have. As discussed in Section 6, the Block Two-Level Erdős–Rényi (BTER) model is good for simulation of graph with this property. However, a Chung–Lu/Stochastic Blockmodel hybrid can allow incorporation of additional attributes, which may be of interest. In terms of uncertainty models, a reasonable model for uncertainty in the context of a dataset can be inferred, as discussed in Sections 5.1 and 5.2. This may be a moot point, however, since the latent graph can be recovered via weighting without considering the specific mechanisms. On the other hand, knowledge of the mechanisms may give an indicator of the reliability of the source. These phenomena are discussed in Section 4.

This page intentionally left blank.

4. LOSS QUANTIFICATION

The purpose of this task was to quantify the loss in performance when the uncertainty mechanisms are applied in the observation process. In particular, the impact on detection performance and parameter estimation are of interest.

Under this task, several experiments were performed to demonstrate both the negative impact of data corruption on detection and estimation algorithms and the improvement in performance that can be gained via fusion of multiple observations.

4.1 ANALYSIS OF UNIFORM EDGE REMOVAL

As a first experiment, the SPG detection framework was applied to an R-MAT background graph, possibly with a small cluster embedded. Before being received by the processing chain, the uniform edge removal model was used to simulate missing data. To simulate model mismatch, it was assumed that the background was generated by a Chung–Lu model rather than the R-MAT model. Detection performance is shown in Figure 3, using an algorithm based on a projection of the graph residuals into two dimensions [1]. While there is a performance loss due to model mismatch, as demonstrated in the figure, this does not account for the additional loss in performance due to the uncertainty mechanism, as shown in Figure 4.

Considering a simpler background model explains the cause of this loss in detection performance. If an Erdős–Renyi background graph were used, all pairs of vertices would have the same probability p of sharing an edge, and thus the residuals matrix of this graph has an eigenvalue distribution that tends (as the number of vertices goes to infinity) to a semicircle with radius proportional to the standard deviation of the entries[1], i.e., $\sqrt{p(1-p)}$. If the graph is sparse, p will be close to zero, and the largest eigenvalues will scale roughly proportionally to $\sqrt{p}$. This is demonstrated in Figure 5, with semicircles of Erdős–Rényi graphs shown on the righthand plot. While the Chung–Lu model allows edge probabilities to vary between different pairs of vertices, there is similar scaling of the maximum eigenvalue with respect

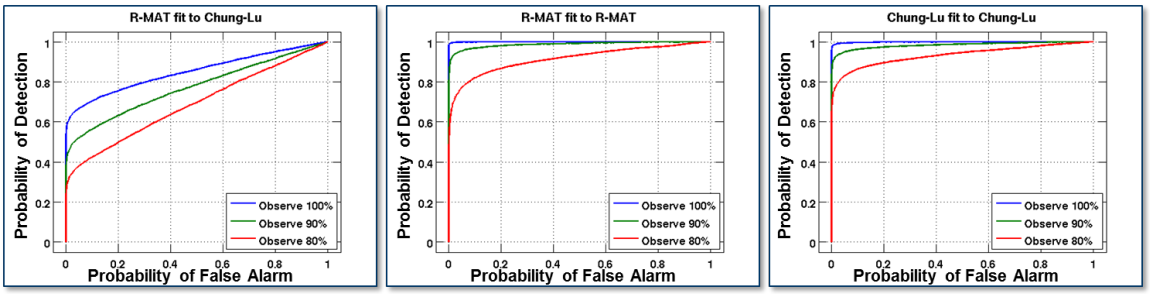

Figure 3. Detection performance with uniform edge removal. When detecting the presence of a cluster embedded into an R-MAT background, fit to a Chung–Lu model, performance is shown on the left. There is a reduction in performance due to model mismatch, as demonstrated by fitting the R-MAT background to an R-MAT model (center) and a Chung–Lu background to a Chung–Lu model (right).

[1] See, e.g., http://mathworld.wolfram.com/WignersSemicircleLaw.html

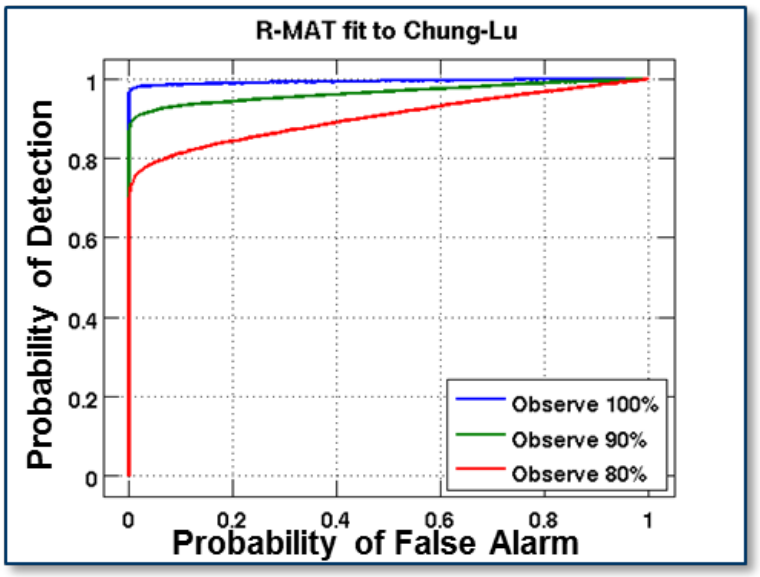

Figure 4. Detection performance with a stronger foreground. When the strength of the foreground increases, detection performance is similar to cases without model mismatch shown in Figure 3.

to the observation probability (lefthand plot). Since the foreground power is characterized by the eigenvalues of its adjacency matrix rather than its modularity matrix, it will scale proportionally to the observation probability, thus causing lower detection performance as its power is reduced more quickly than that of the background.

4.2 FUSION OF MULTIPLE OBSERVATIONS

While uncertainty in data can quite negatively impact detection performance, fusing multiple corrupted observations may allow some of this loss to be regained. As notionally depicted in Figure 6, it is desirable for the framework to accept several observations, possibly with significantly different uncertainty mechanisms, and analyze the resulting data with an increase in detection and inference ability.

Fusion of multiple observations was considered in two categories. In one case, distributions of the observations are assumed and the expected loss (e.g., number of edge errors) is minimized. In the other category, no prior knowledge of the distributions of the observations is assumed, but each observation has a weight corresponding to its "reliability." In simple cases, such as an Erdős–Rényi graph with uniform probability of error, the former kind of Bayesian analysis may be tractable, as shown in Figure 7, but it can become much more complicated with more sophisticated models for error.

The ability to recover detection performance with a weighting scheme is demonstrated in the following set of simulations. The simulation setup is similar to that in Section 4.1. As shown in Figure 8, each error mechanism has a unique impact on detection performance. Each mechanism was set to create 20% edge errors, i.e., the number of false edges plus missing

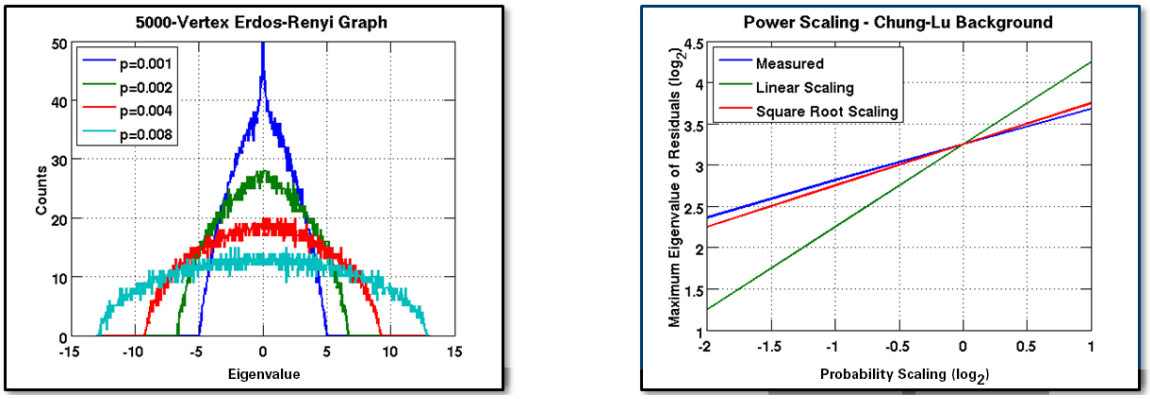

Figure 5. Background power reduction with missing data. For an Erdős–Rényi background, the distribution of eigenvalues follows a semicircle distribution which, for sparse graphs, has a radius proportional to the square root of the edge probability (left). While a Chung–Lu graph has more complicated structure, its maximum eigenvalues also tend to scale proportionally to the square root of the observation probability (right).

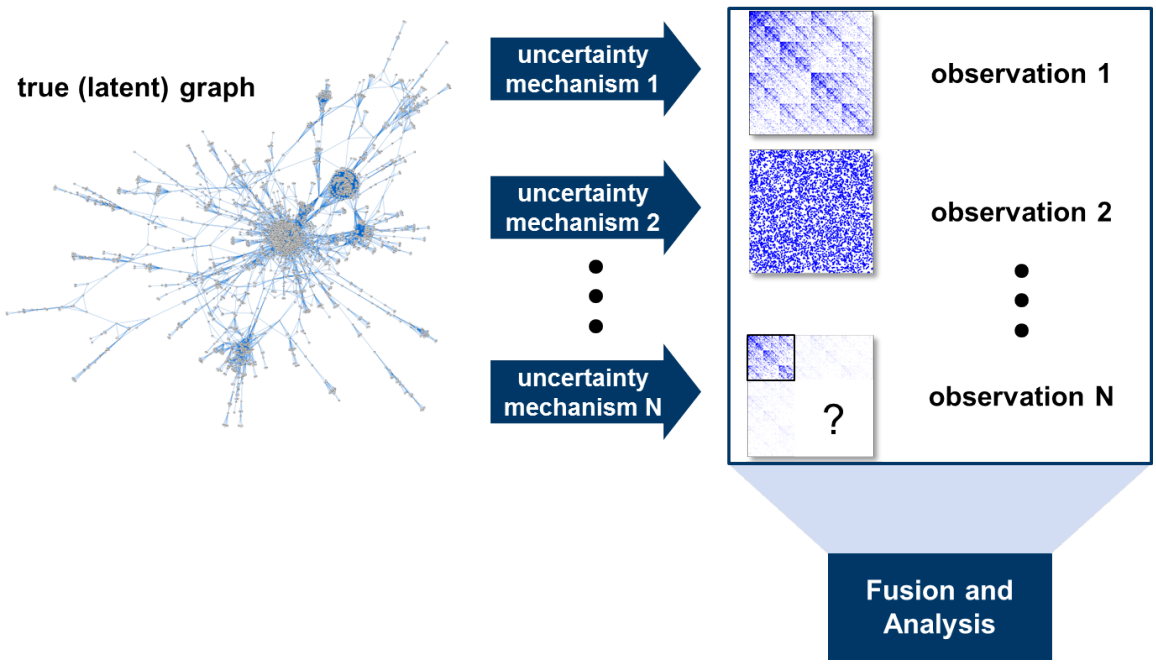

Figure 6. Notional diagram of multi-observation fusion. A desirable outcome of this study is a demonstrated ability to fuse multiple corrupted observations and provide an increase in detection performance.

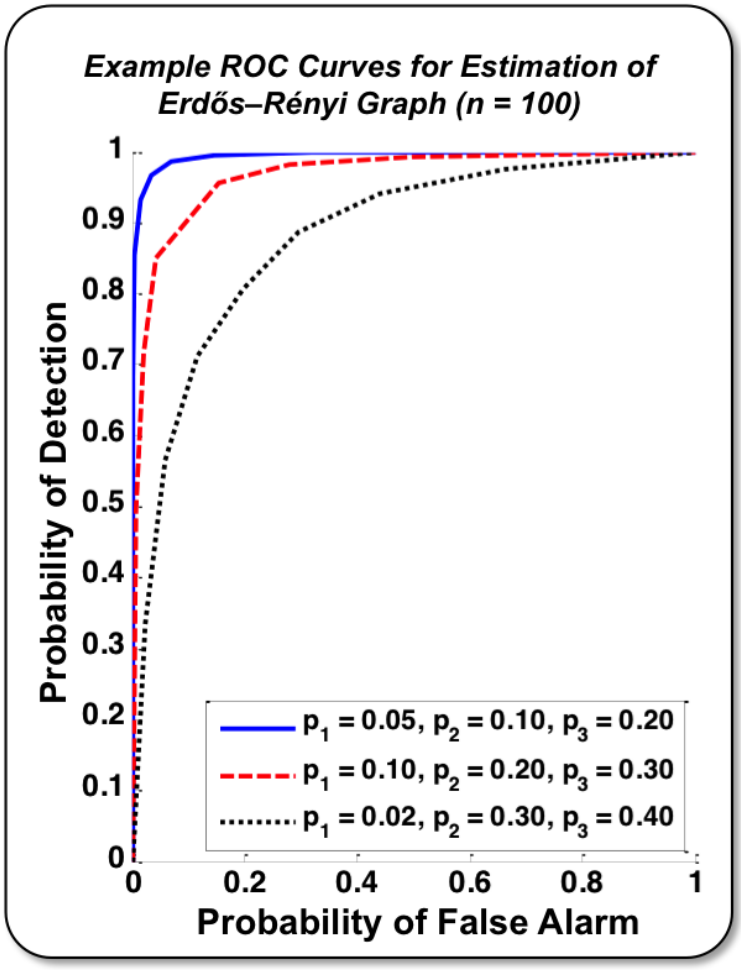

Figure 7. Recovery of an Erdős–Rényi random graph with uniform edge errors. The simple structure of the problem yields a decision rule based on which edges exist in each of the three observations. Edge error probabilities are shown in the legend.

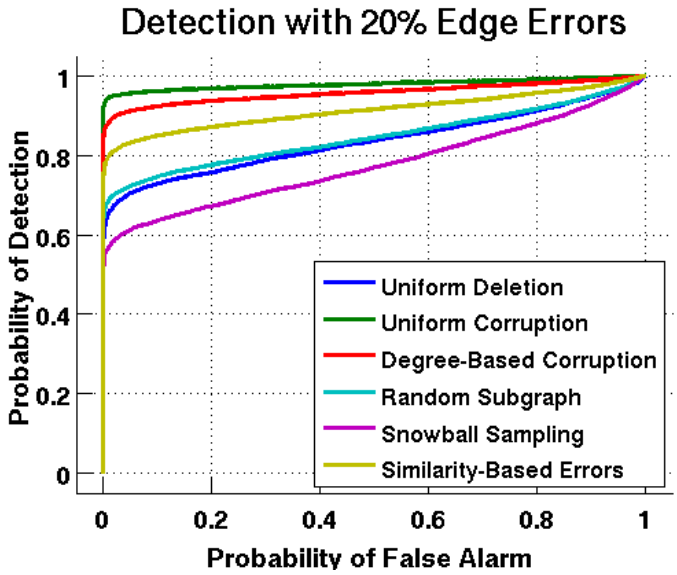

Figure 8. A comparison of the impact on detection performance in simulation using each uncertainty mechanism. Each mechanism was tuned to create 20% edge errors in expectation.

edges is 20% of the total number of true edges, in expectation. Uniform corruption leaves performance roughly equal to uncorrupted detection performance, since the errors do not correlate and are less strong than the embedded cluster. Degree-biased edge errors cause some loss in performance, since they somewhat correlate with the background noise. Similarity-based errors degrade performance slightly more, and random edge and vertex removal seem to cause quite similar reductions in performance. The mechanism that causes the greatest reduction in performance is snowball sampling.

Observations from four mechanisms (uniform deletion, degree-biased edge errors, snowball sampling and similarity-based errors) were fused in an attempt to recover detection performance. Using a simple fusion method in which all edges are considered only matched performance of the highest-performing observation (though there is often an increase in performance if only mechanisms that remove, but do not add, edges are considered). Weighting each observed graph in accordance to its individual performance, however, allows recovery of performance on the latent graph itself, as shown in Figure 9.

Additional cases were considered in which less redundant data were available. In these scenarios, a graph corrupted via degree-based corruption with an average edge error of 50% was fused with another graph with 20% edge errors, either with similarity-based errors or uniform edge removal. As demonstrated in Figure 10, the case of fusion with similarity-based errors recovers equivalent performance with the latent graph, while fusion with a graph with uniform edge removal, which has a much more substantial impact on detection performance, yields a slightly lower probability of detection. Thus, even using relatively little data, the performance lost to various error mechanisms can be regained through the simple weighting scheme employed here.

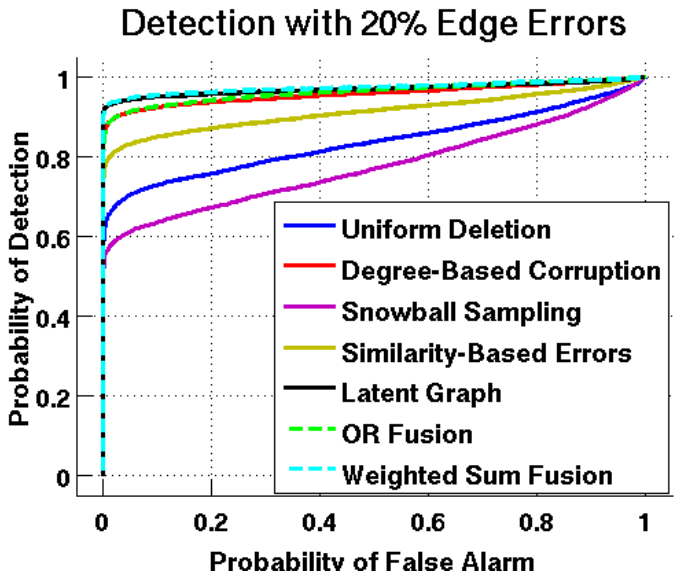

Figure 9. Detection performance when fusing four observations via a weighted combination. Simply taking all vertices across observations provides the same performance as degree-biased corruption, but a weighted combination of all observations recovers detection performance with no uncertainty mechanism in place.

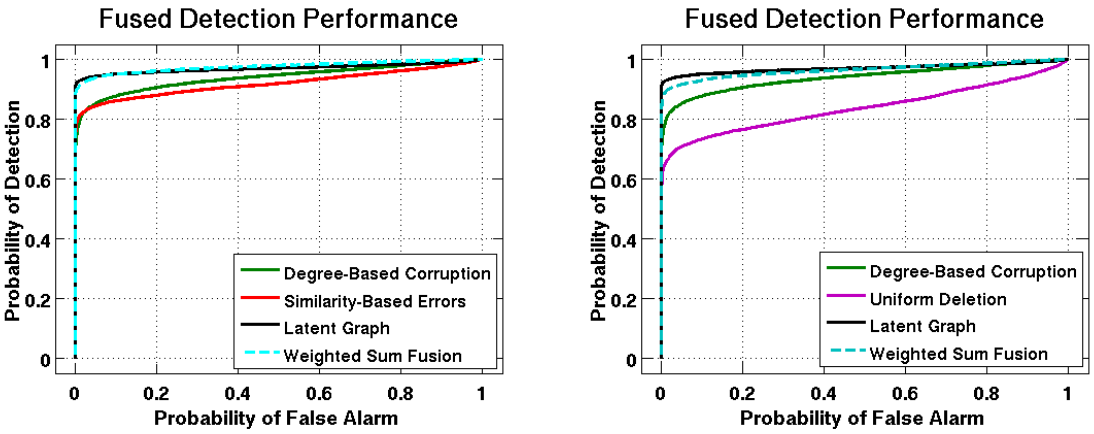

Figure 10. More difficult cases of multi-observation fusion. A graph with greater degree-based corruption is fused with a graph with similarity-based errors (left) and uniform edge removal (right). Fusion with the graph with similarity-based errors recovers equal detection performance to the latent graph, while fusion with the graph with deleted edges yields a small gap in performance.

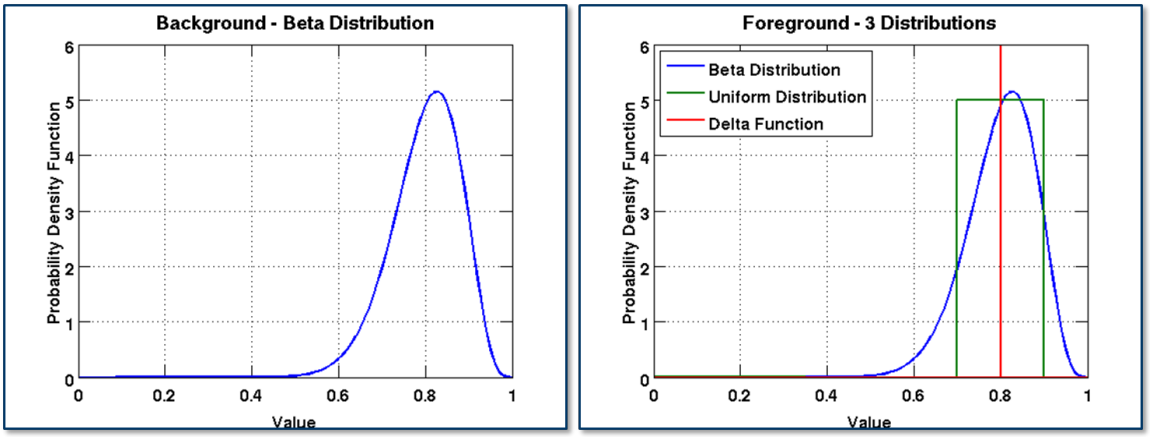

Figure 11. Distributions of edge weights for simulation. The edge weights for the background follow a beta distribution (left), while the foreground weights may have either a beta, uniform, or delta distribution (right). In all cases, the expected edge weight is 0.8.

4.3 IMPACT OF THRESHOLDING EDGE WEIGHTS

One potential concern when considering graphs with uncertainty is how to threshold the graph, i.e., how much confidence there must be in the existence of an edge before considering the edge to exist. An experiment to that effect was performed, with results indicating that a weighted, rather than thresholded, graph should be used in the analysis process. The same background and foreground as in Section 4.1 is used, and edges are given a weight according to the beta distribution shown in Figure 11. The foreground graph is given weights from three distributions with the same expected value, also shown in the figure. The change in detection performance for the different edges' weighting schemes is displayed in Figure 12. There is a small but appreciable difference in detection performance when including the edge weights, up to a 5% increase in probability of detection for a given false alarm rate.

When a graph with weighted edges has those weights thresholded, on the other hand, the difference in performance is much more significant, as shown in Figure 13. When the foreground and background come from the same distribution, there is a substantial decrease in performance. By the same logic as the case explored in Section 4.1, this is because the foreground power is being reduced at a greater rate than the background power. When the distributions differ, weighting can actually be beneficial to performance, as also shown in the figure. In the case where the threshold is set below the lower bound of the support of the foreground weight distribution, there can be a boost in performance, as background activity will be reduced. This is unlikely to be known in practice, however, and an error in such a judgement call may have a substantial negative impact on the ability to detect anomalous subgraphs.

4.4 CHALLENGES AND RECOMMENDATIONS FOR FUTURE WORK

One significant challenge in loss quantification is evaluation of the impact on parameter estimation. Uniform edge removal has the advantage of only removing data, and removing

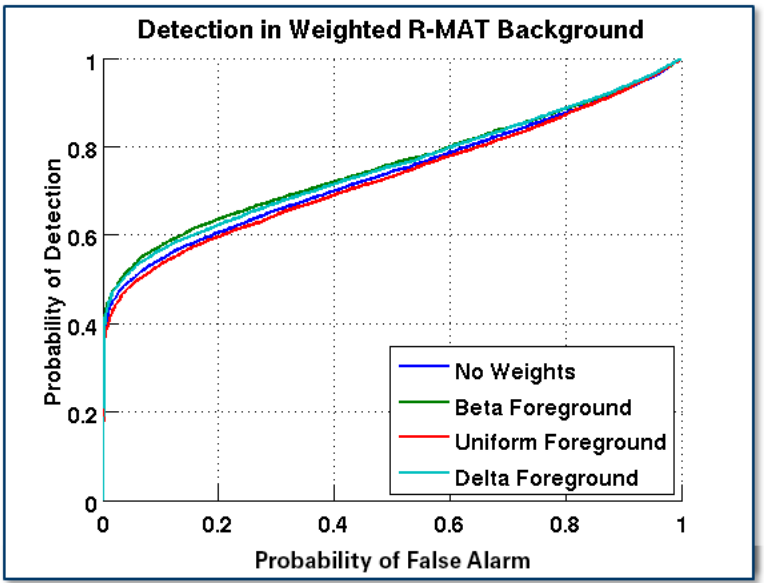

Figure 12. Detection performance with weighted edges.

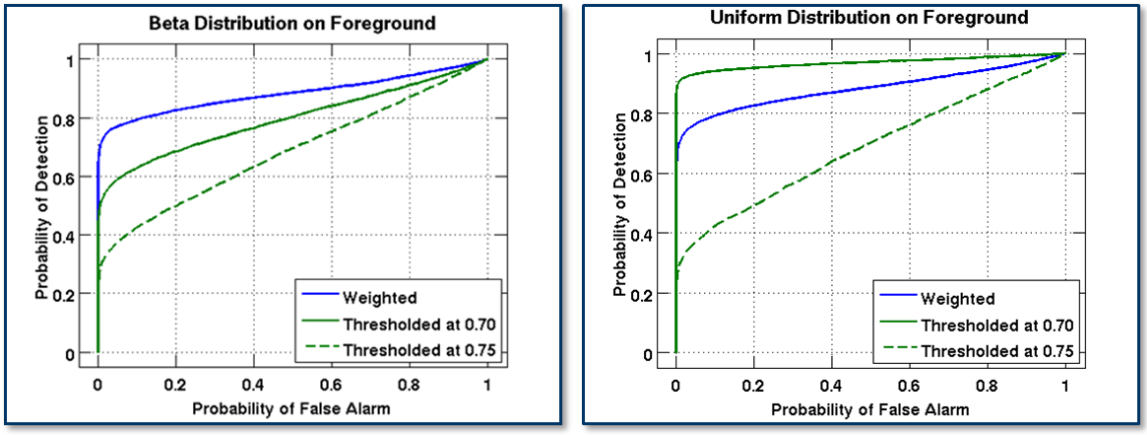

Figure 13. Detection performance when thresholding weighted edges.

data equally across the graph. Thus, the impact is that edge deletion is commensurate with the loss of data due to the missing edges. This is also true when only a subset of vertices are observable. However, for random corruption models, the uncertainty mechanism presents a challenge for the estimation procedure, requiring iterative estimation procedures that may not scale to sizes of interest.

Given the difficulty in estimating parameters in the presence of uncertainty mechanisms, and the ability to detect in weighted graphs (and in particular those obtained through multi-observation fusion), the most appropriate algorithms when operating on extremely large graphs should not rely on detailed knowledge of the uncertainty mechanism, but rather fuse observations via a simple weighting scheme[2], which has been shown to be effective in this study.

In practice, edges are often chosen or disregarded depending on the confidence of an analyst in the veracity of the data. As discussed in Section 4.3, however, this has the potential to drastically degrade detection performance if there is not accurate prior knowledge of the distribution of edge weights (or, in some cases, even if those distributions are known perfectly). This suggests that evidence of the presence of an edge should be weighted by confidence in the observations (similarly to the multi-observation fusion example), to avoid disregarding information that could substantially improve signal detectability.

[2] The weighting scheme used here is to apply the logistic function $f(x) = 1/(1+e^{-x})$ to the sum of weights corresponding to existence of edges from different observations. That is, if an edge between i and j exists in observation o (i.e., $a^o_{ij} = 1$), then weight w_o will be included in the sum. Thus, an edge in the resulting graph has weight $f\left(\sum_o a^o_{ij} w_o\right)$ if it exists in any of the observations.

This page intentionally left blank.

5. DATA ANALYSIS AND MODEL REFINEMENT

The purpose of this task was to analyze real datasets to determine the applicability of the uncertainty mechanisms to real, dynamic graph datasets. This involves considering known errors in the data, and applying the uncertainty mechanisms to the data, treating the observed graph as "truth," and determining the impact on detection and estimation performance.

Under this task, errors in the Web of Science data and the web proxy logs were analyzed to understand the way in which uncertainty manifests itself in real data. These errors informed some of the models presented in Section 3. In addition, uncertainty mechanisms were applied to the Web of Science data to determine the impact of these models on real datasets.

5.1 ERRORS IN WEB OF SCIENCE

Upon initial investigation in the Web of Science database, one particularly egregious error that came up relatively frequently was the citation of documents several decades in the future. Considering these edges in particular allowed insight into uncertainty and error mechanisms in this real dataset. Three distinct forms of error were observed in the data. One mechanism involved several documents in 1997 being cited by documents in the 1970s. While this provided an interesting case study of error correction using a few simple D4M (Dynamic Distributed Dimensional Data Model) commands, there was not much useful information that could be gleaned from the context of these errors. The document identifiers were not similar, nor was any other vertex metadata. Another common error is two vertices that have the same document identifier. A mechanism like this can be modeled similarly to the similarity-based uncertainty mechanism, with simple random edge rerouting. Finally, one error category inspired the similarity-based uncertainty mechanism described in Section 3. This involved records pointing to other database entries that matched (1) the last two digits of the year, (2) the page number, (3) the journal volume, and (4) the first four letters of the first author's last name.

5.2 ERRORS IN WEB PROXY LOGS

The most apparent source of error in the web proxy logs is occasional loss of connectivity. At various points in the available data, there is no traffic recorded for several minutes at a time. The result of this is an uncertainty mechanism like uniform edge deletion, assuming that the connections made in the time periods in which data are missing is representative of the overall traffic patterns.

5.3 UNCERTAINTY MECHANISMS APPLIED TO WEB OF SCIENCE

In the previous study, some interesting emergent behavior was found surrounding the year 1976 in the Web of Science citation graph, involving analytical chemistry papers receiving significant cross-subject citations. Uncertainty mechanisms were applied to the citation graph around this period in time, to determine the impact of these mechanisms on a real dataset. In particular, uniform edge removal and uniform edge error were applied to the data. In the case of

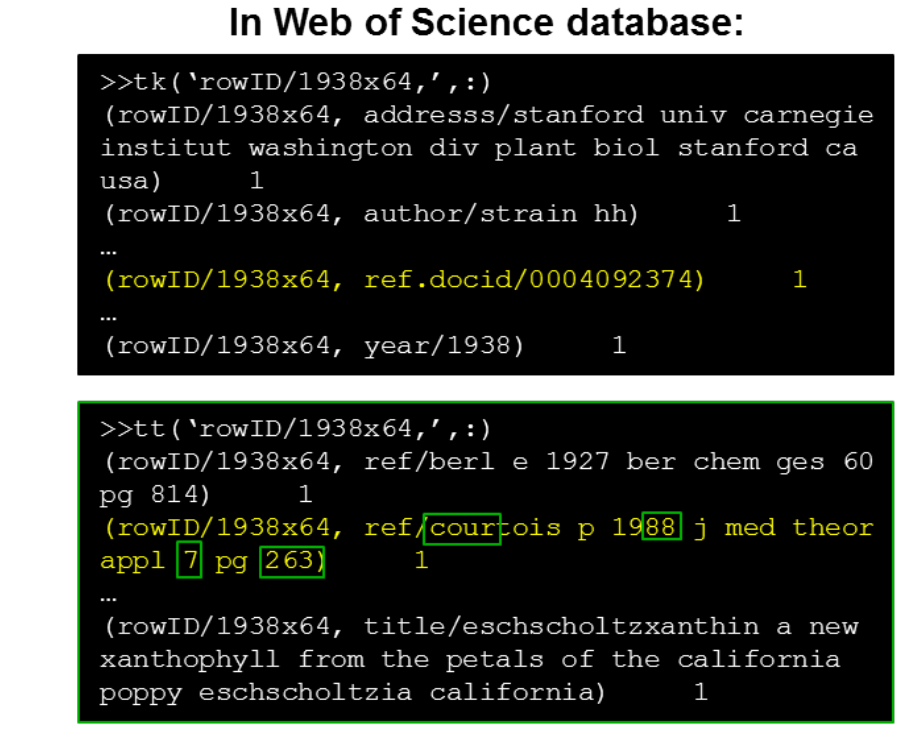

Figure 14. Example error in Web of Science database.

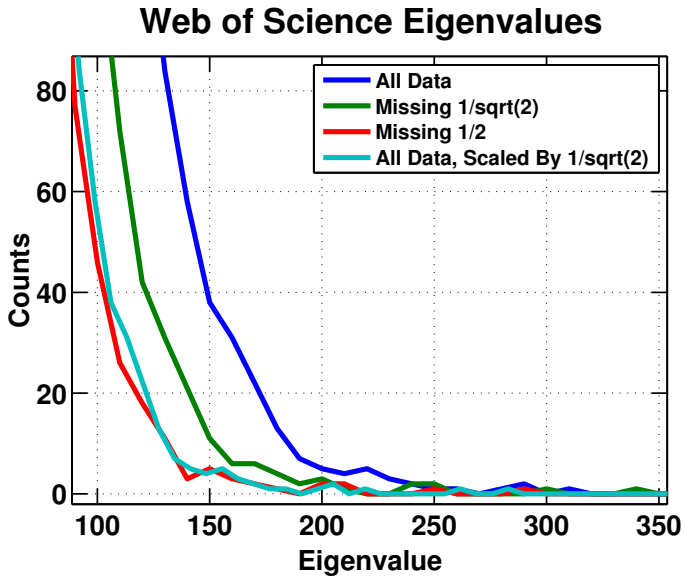

Figure 15. Singular values of Web of Science citation graph with randomly removed edges. Note that the "radius" of the distribution when half of the edges are removed is reduced by approximately a factor $\sqrt{2}$, as in an Erdős–Rényi random graph.

uniform edge errors, no significant difference was seen in the resulting spectral space, since the random error edges were quite weak in comparison to the actual background. When random edges were removed, while the subgraph was still detectable due to its strength, there was a similar scaling of the eigenvalue distribution as was seen in the Erdős–Rényi case. As shown in Figure 15, the distribution of the singular values of the integrated residuals matrix when half of all edges are removed contracts by approximately a factor of $\frac{1}{\sqrt{2}}$. This demonstrates a case in which missing data impacts the background power just as it does in a simple simulated background.

Another experiment was run in which the uniform corruption mechanism was applied to a subset of the Web of Science, and the task was to recover the original graph. Three observations were made: one with 0.1 errors per vertex, one with 1 error per vertex, and one with 10 errors per vertex. Results are shown in Table 2. Empirical risk minimization assumes all observations are equally good, and thus provides an estimate close to the one with many errors per vertex. By weighting the estimates, a significant reduction in the number of errors is achieved, made even smaller by prior knowledge of the number of true edges. This constrained version provides exactly the same performance as the Bayesian approach that assumes an Erdős–Rényi distribution.

5.4 CHALLENGES AND RECOMMENDATIONS FOR FUTURE WORK

The greatest challenge in analyzing uncertainty in real data is the lack of ground truth. To understand the causes of error in real data requires knowledge of the latent graph, which is typically not available, and requires using portions of the data obviously in error, as done in this section. This can lead to interesting models for graph errors, but there is no guarantee

Method	Average Errors (Web of Science)	Average Errors (Erdős–Rényi)
Empirical risk	162,643	162,784
Weighted empirical risk	5,374	5,417
Constrained weighted empirical risk	6	0
Bayesian (Erdős–Rényi prior)	6	0

TABLE 2

Recovery of the Web of Science citation graph from multiple corrupted observations.

that all sources of uncertainty have been covered. This is another reason to use a weighting scheme based on source reliability, as suggested in Section 4.4. In addition, the empirical results in Section 5.3 demonstrating that weighting observed graphs enables equal performance to a Bayesian technique for recovery of a real graph suggest that such a technique would be broadly applicable.

This page intentionally left blank.

6. DATA FRAMEWORK DEVELOPMENT

The purpose of this task was to build a set of data characteristics and a framework for comparison of graph data. This will ensure that proposed algorithms are applicable in a wide variety of scenarios.

Under this task, MIT LL proposed the list of graph characteristics shown in Table 3. This covers a standard set of graph characteristics, in particular focusing on those that can be computed for extremely large graphs (which is why, e.g., betweenness centrality is not included). A diverse set of large network datasets were characterized in the context of these features. In addition, the Block Two-Level Erdős–Rényi (BTER) generator was used to simulate networks with similar degree and clustering coefficient distributions [14].

6.1 NETWORK DATASETS

In addition to the Web of Science and web proxy logs, many datasets from publicly available repositories were analyzed. Data from the Stanford Network Analysis Project (SNAP)

Characteristic/Statistic	Relevant Questions
Vertex categories	Are there multiple distinct types of vertices? Is the graph bipartite (or multi-partite)?
Weights	Is there a natural scale regarding the edges? If there are non-binary values, what are the units?
Update rate	How quickly do the data (both vertex and edge sets) change?
Relationship types	Are there distinct relationships between the same pairs of vertices? How is each relationship characterized by the above features?
Size	How many entities and relationships are in the data, per time?
Degree distribution	How many of the vertices have low/high/intermediate degree? How quickly does the number of vertices with a given degree decrease as degree increases? Does the degree distribution change over time?
Clustering coefficient distribution	How many complete triangles are in a typical node's neighborhood? Does this tend to change with degree (e.g., do low-degree vertices have a higher clustering coefficient)?
Eigencentrality/PageRank	How influential are nodes according to these centrality metrics?
Size of connected components	How many vertices can be reached by tracing relationships?
Average distance (eff. diameter)*	How many hops between the average pair of vertices?

*May not scale to extremely large graphs

TABLE 3

Characteristics of graph data.

large dataset collection[3] and the repository of the *Laboratorio di Algoritmica del Web* (LAW – Laboratory for Web Algorithmics)[4] were downloaded, comprising the following 13 graphs:

- US patent citation network
- Amazon product similarity links
- Enron email network
- Wikipedia voting network
- Road networks for California, Pennsylvania, and Texas
- LiveJournal and Friendster networks
- Gnutella peer-to-peer network
- Autonomous system traceroute
- Traces of .eu and .cnr domains from 2005 and 2000, respectively

Global statistics for these graphs, as well as different time resolutions for the Web of Science and web proxy graphs and four instantiations of the BTER model, are shown in Table 4[5]. Among the global statistics, most have known correlations (number of edges scaling slightly superlinearly with number of vertices, number of triangles scaling similarly to number of edges, etc.). The biggest outliers are the road networks, due to their extremely high diameters. Among the other datasets, considering their degree-based statistics (plotted in Appendix A) shows that clustering coefficient always drops precipitously with degree, and PageRank and eigencentrality often linearly increase with degree. Four notable exceptions to this trend are the autonomous system, Friendster, Wikipedia, and patent citation graphs, in which PageRank does not reliably increase with degree. Also, in several cases, such as LiveJournal, eigenvector centrality appears "bimodal," i.e., a given vertex seems to be on one of two lines dictating eigencentrality with respect to degree. Generating graphs from the BTER model matches many of the characteristics of the real data, including varying dependencies of eigencentrality on degree. In particular, the BTER generated graph fit to the Enron email dataset has a characteristic profile strikingly similar to the web proxy graphs.

6.2 SOFTWARE FOR PARALLEL ANALYSIS

Over the course of the first VLG study [5], MIT LL considered uncued anomaly detection in large, dynamic graphs with attributes by analyzing the spectral properties of the graph residuals (i.e., the difference between the observed graph and its expected value under an assumed model) [13, 15, 16]. In addition to analyzing two real datasets, synthetic data was

[3] Available at http://snap.stanford.edu/data/index.html.
[4] Available at http://law.di.unimi.it/datasets.php.
[5] "OVERFLOW" for the number of triangles in the is due to lack of precision in the software used to compute the statistics. According to the SNAP website, the number of triangles in the graph is 4173724142.

Dataset	# Vertices	# Edges	Effective Diameter	Diameter	# Triangles	Avg. Clust. Coeff.
Patent Citation	3774768	16518948	9.181146	18	7515023	0.08
Enron	36692	367662	4.72041	12	7515023	0.5
LiveJournal	4847571	68993773	6.870505	18	285730264	0.27
Friendster	65608366	1806067135	5.858562	23	OVERFLOW	0.162295
Gnutella	6301	20777	8.471423	20	2383	0.01
Autonomous System	1696415	11095298	9.337051	36	28769868	0.26
.eu Domain	862664	19235140	7.883786	19	202170577	0.60815
.cnr Domain	325557	3216152	14.765241	30	20977629	0.452944
Wikipedia Voting	7115	103689	3.971075	10	608389	0.14
Amazon Product Similarity	735323	5158388	9.972961	23	4464791	0.355271
California Road	1965206	5533241	500.761632	851	120676	0.05
Pennsylvania Road	1088092	3083796	533.368422	786	67150	0.05
Texas Road	1379917	3843320	667.416515	1058	82869	0.05
BTER LiveJournal	4762974	86783914	5.807724	14	527840253	0.57
BTER Autonomous System	1679032	21392478	4.989197	13	200397577	0.69
BTER Wikipedia	6895	99880	3.919973	9	597207	0.145521
BTER Enron	35798	176492	4.780367	9	1112672	0.582943
Web of Science: 1990	3301147	7787200	11.141949	25	565526	0.0222
Web of Science: 2000	6179522	16413449	9.973632	26	1232648	0.020607
Web of Science: '86-'90	6781053	36554823	8.51555	23	24765408	0.085228
Web of Science: '96-2000	11393928	70175362	7.90054	22	49914742	0.07915
Web of Science: '81-2000	16548581	183409708	6.609184	20	299670458	0.129466
Web Proxy: One Minute	3429	8875	5.557352	9	0	0
Web Proxy: One Hour	19742	226156	3.850693	8	0	0
Web Proxy: One Day	56132	1423536	3.829959	7	0	0

TABLE 4

Statistics of real networks.

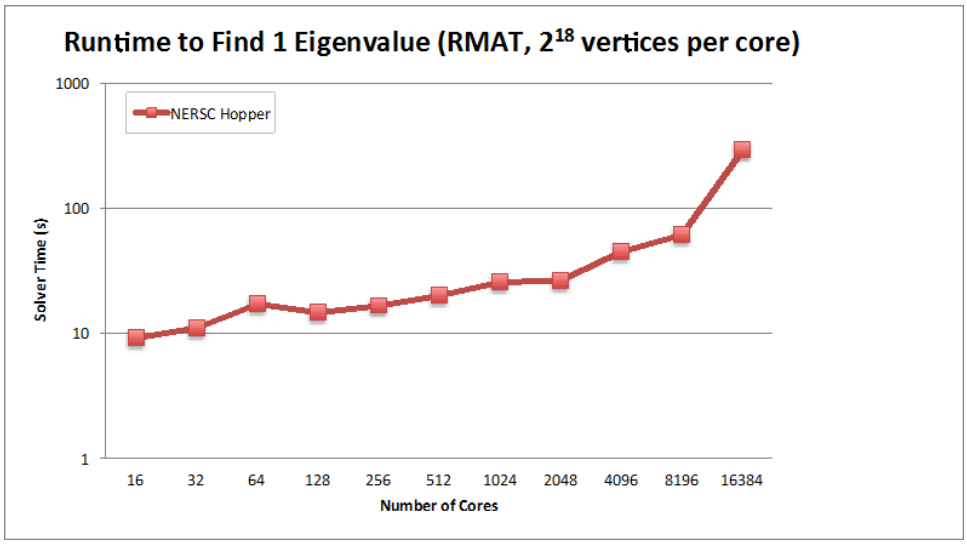

Figure 16. Weak scaling of parallel eigendecomposition. The rightmost point in the plot demonstrates a 4-billion-vertex graph being processed in less than 5 minutes.

generated to test the algorithms at greater scale. The key algorithm in the SPG processing chain—the partial eigendecomposition of a sparse matrix minus its low-rank expected value—was implemented using the Scalable Library for Eigenvalue Problem Computations (SLEPc) for parallel eigenvalue computation [17]. Using this method, the principal eigenvector of a billion-vertex graph's modularity matrix was computed on 64 commodity compute cores. Under the present study, the Anasazi library was used, and was run on the supercomputer Hopper at the National Energy Research Scientific Computing (NERSC) Center. Results for weak scaling (2^{18} vertices per core) are shown in Figure 16. Running time per processor increases fairly slowly, until over 2000 processors are used. Despite poor scaling with large numbers of processors, this demonstrates the ability to compute the principal eigenvector of a four-billion vertex graph's residuals (modularity) matrix in under five minutes. Strong scaling results (8M vertices) are shown in Figure 17, also demonstrating a decrease in performance per core for large numbers of cores.

6.3 CHALLENGES AND RECOMMENDATIONS FOR FUTURE WORK

One noteworthy observation of the real data is that, while the road networks are substantially different from the other graphs in their diameters, the variations in statistics for the other graphs do not separate the graphs into qualitative classes. For example, the autonomous system graph and the patent citation graph have similar clustering coefficient distributions despite being fundamentally different in terms of their rate of change, while the autonomous system is substantially different from the .eu and .cnr domains, despite all being internet-based graphs. Thus, the statistics considered here do not seem to provide a basis for discriminating between various graph types, e.g., distinguishing fast, transaction-based data from slower-moving social networks.

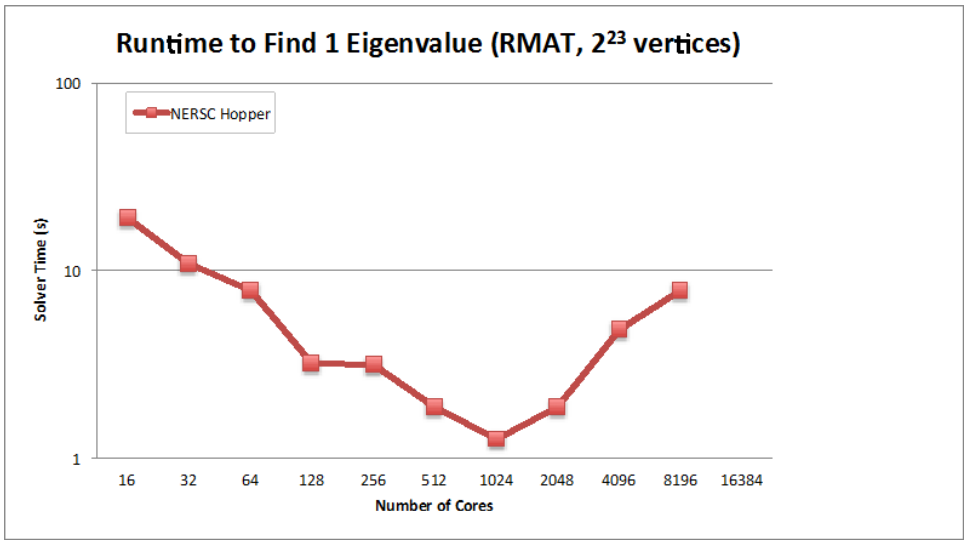

Figure 17. Strong scaling of parallel eigendecomposition. Computation time per core increases after 1024 cores, likely due to communication overhead.

To ensure that data analytics are applicable to a variety of graph data, a framework must be tested on data from a diverse set of sources. Apart from the road networks, the real networks considered in this study have characteristics that are by and large similar. Thus, to cover the space of graphs of interest, representative graph datasets should be chosen that are diverse in the areas of greatest variance: the slope of the degree distribution, the average clustering coefficient, and the dependence of eigencentrality and PageRank on degree. Selecting datasets that are diverse in these key characteristics, as well as in terms of the rate of change, will ensure that developed algorithms are applicable to a significant cross section of graph data.

Simulation of large graphs with characteristics similar to those in real networks is necessary for evaluation with ground truth, and the BTER model has proven to capture many of these properties. One in particular is lacking: it always generated graphs where PageRank is linearly dependent on degree. Despite this, BTER provides a level of accuracy that is missing in other current generators, and there is new open source software for fitting to real data and generation on parallel machines[6].

Computational performance using the NERSC machine was quite favorable in comparison to LLGrid, largely due to the optimized communication network. For parallel computation, even on a network highly tuned for interconnected computation, there is a loss in performance when using a large number of cores. This is, in part, because one-dimensional partitioning is being used, where each processor has a block of rows. This requires all-to-all communication between iterations of the eigensolver, which is the likely bottleneck. Changing to a two-dimensional partitioning scheme, which is allowed in Anasazi but not SLEPc, should enable better scaling performance.

In addition to computational scaling issues, a loss in precision has been noticed when multiplying by the rank-1 expected value in the modularity matrix. Multiplying by this term

[6] Available at http://www.sandia.gov/~tgkolda/feastpack/.

gave unstable performance in terms of convergence, which was circumvented for the time being by using several averaged background models as the expected value, still resulting in a sparse matrix, but with more nonzeros. This problem would best be resolved by using extended precision arithmetic, for which there are publicly available software packages. This solution, however, will significantly increase the algorithm's memory footprint.

7. SUMMARY

This report documents research into the effect of uncertainty on detection and inference in very large graphs. Several models for uncertainty are proposed, including some from the open literature and some based on experience with real data, including the Thomson Reuters Web of Science database and a set of web proxy logs. The impact of these uncertainty mechanisms on detection performance is evaluated, and a simple weighting scheme based on the impact that a mechanism has on detection performance is shown to enable the same performance as when no uncertainty mechanism is applied. It is also shown that real datasets have errors that stem from uncertainty mechanisms similar to the proposed models, and that some similar phenomena occur when applying the mechanisms to real data as when applying to simple random graphs. Finally, a set of graph features for characterizing very large graphs is proposed, and several large network datasets are analyzed in the context of this feature space. The Block Two-Level Erdős–Rényi model is shown to also have features quite similar to those in real data.

An ongoing follow-on study aims to determine hardware-centered issues with performing this sort of residuals analysis on very large graphs. Under the present study, it is demonstrated that a four-billion vertex graph can be processed in under five minutes using a state-of-the-art supercomputing system. To achieve the ability to process terascale graphs in a similar time frame, the inefficiencies of current computing hardware for processing large, sparse datasets must be addressed.

This page intentionally left blank.

APPENDIX A: CHARACTERISTICS OF GRAPH DATASETS

This appendix includes all vertex-wise statistics of the real graphs analyzed in Section 6. Each of the figures includes the degree distribution, and, with respect to degree, the clustering coefficient, eigencentrality, and PageRank of the vertices. In addition to showing the clustering coefficient, eigencentrality, and PageRank for each individual vertex in blue, the average value for a vertex of the specified degree is shown in black. Simulations using the BTER model are included based on the degree distribution of the autonomous system graph (Figure A.2) and the LiveJournal graph (Figure A.17) in Figures A.3 and A.18, respectively, as well as those fit using an eye-calibrated clustering coefficient distribution for the Enron graph (Figure A.6) and the Wikipedia graph (Figure A.19) in Figures A.7 and A.20. Also, it should be noted that the clustering coefficient is always zero for the web proxy data (Figures A.11, A.12, and A.13), since the graphs are bipartite.

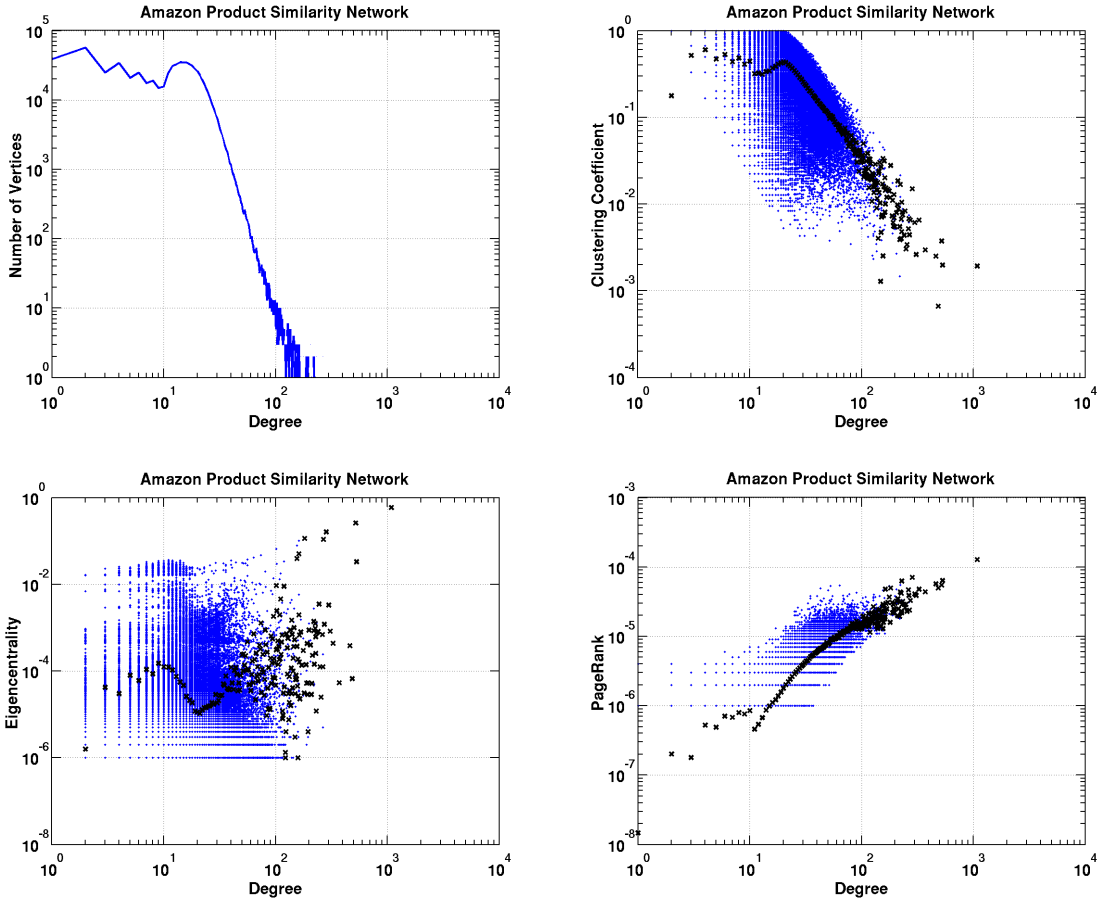

Figure A.1. Vertex statistics of the Amazon product similarity dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

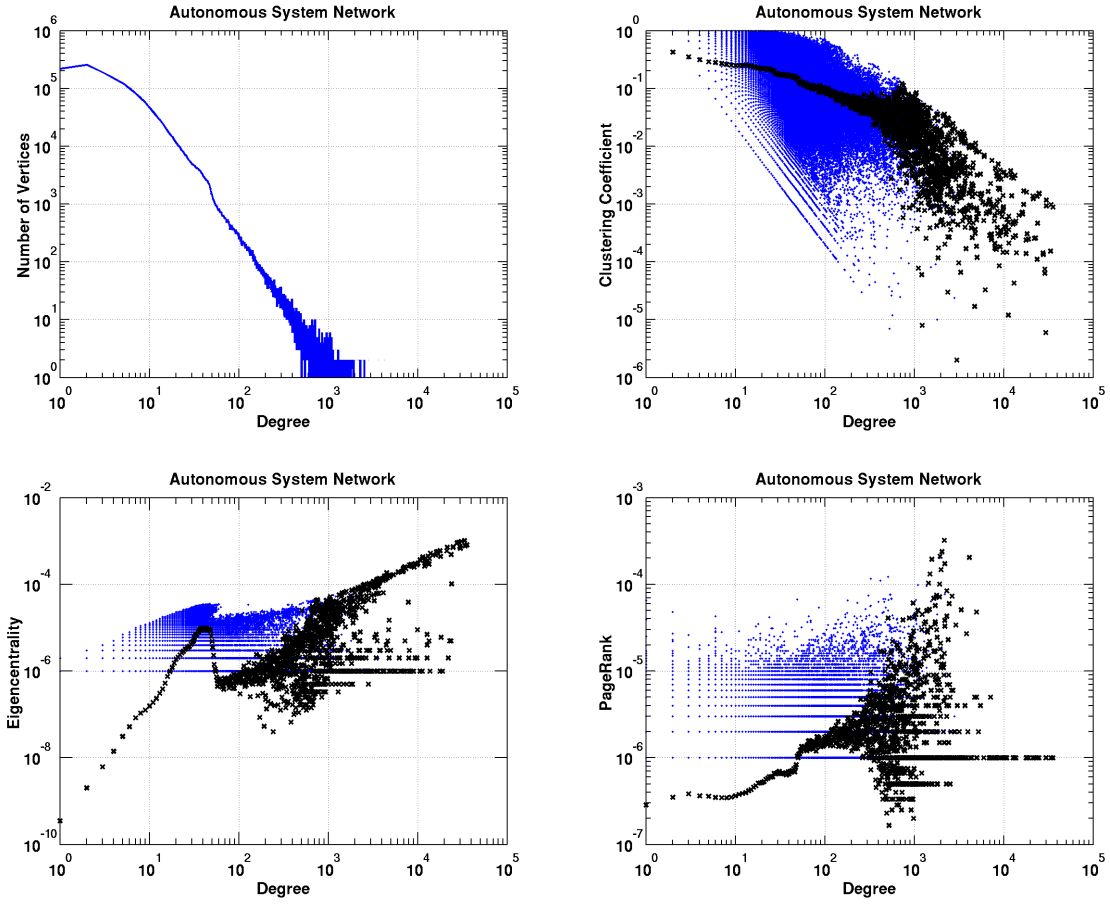

Figure A.2. Vertex statistics of the Skitter autonomous system dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

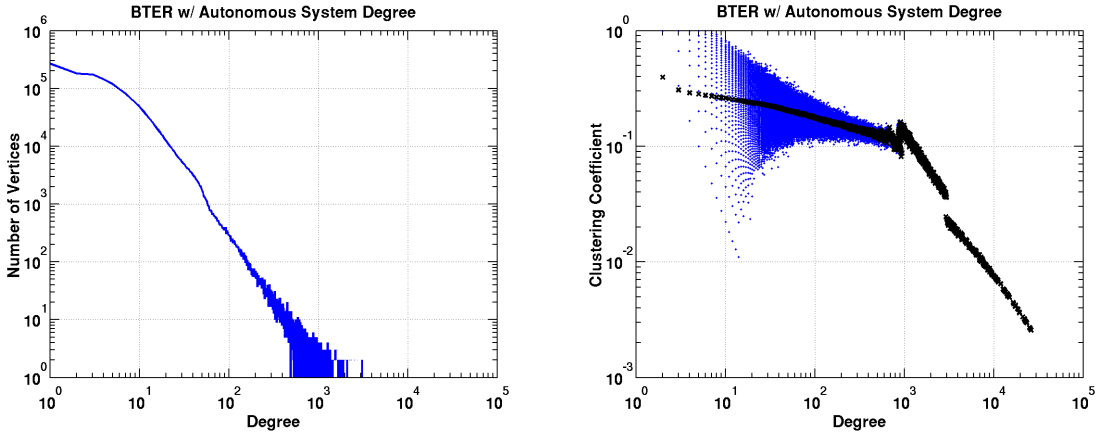

Figure A.3. Vertex statistics of the BTER simulation (with Skitter degree distribution) dataset: degree (left) and clustering coefficient (right). Eigencentrality and PageRank were not computed at the time of completion of this report.

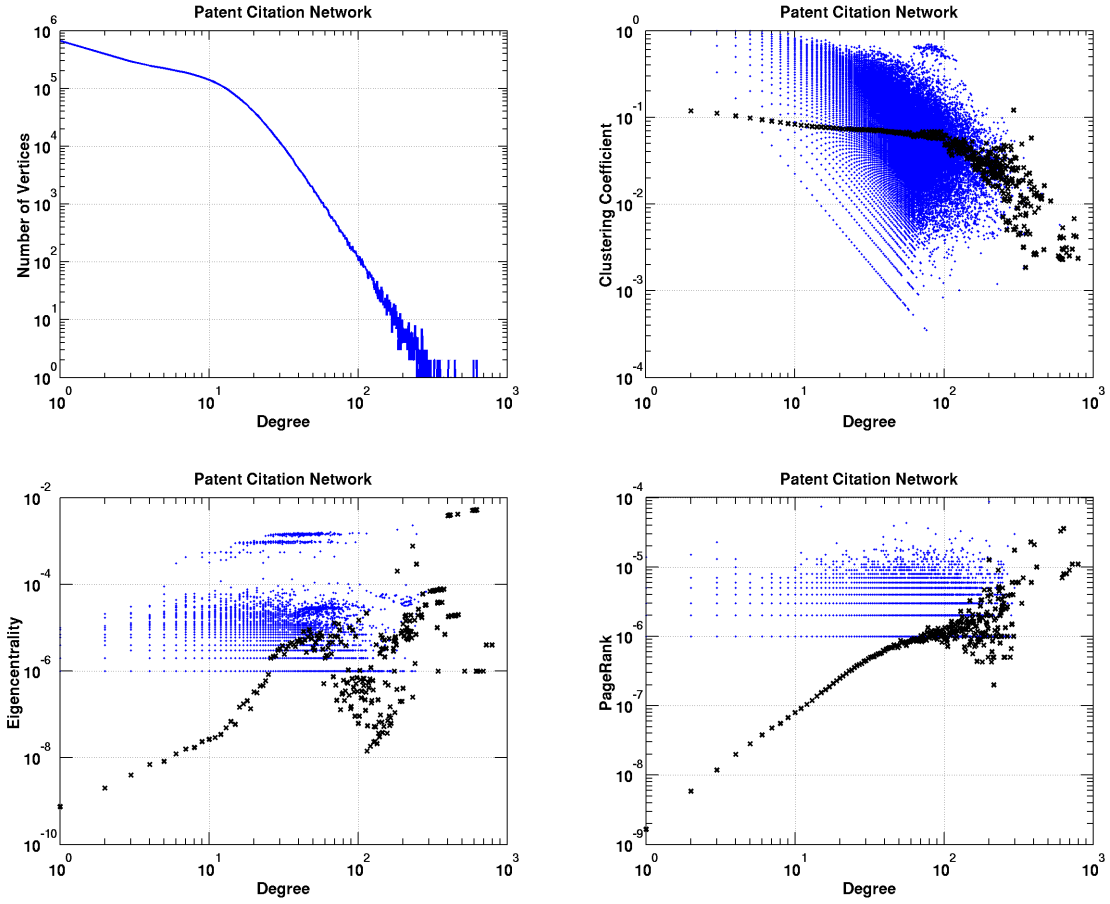

Figure A.4. Vertex statistics of the patent citation dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

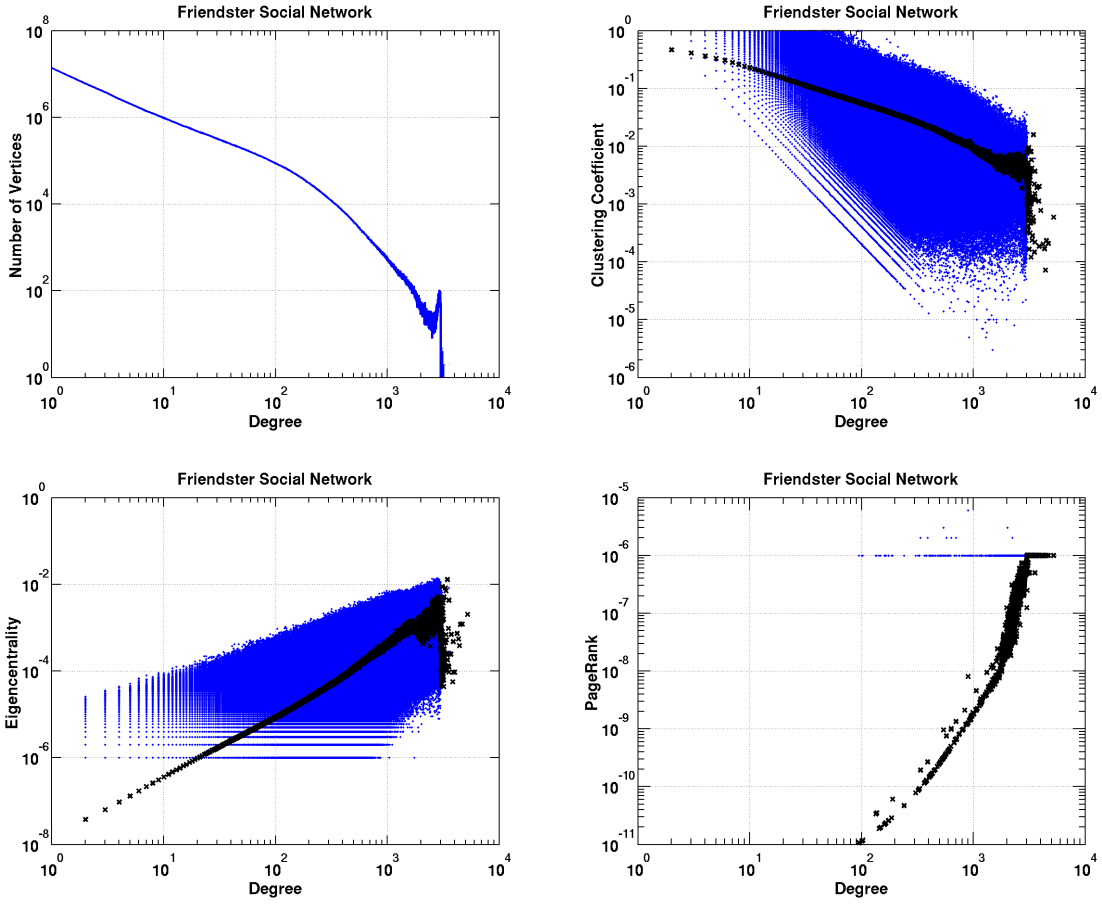

Figure A.5. Vertex statistics of the Friendster social network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

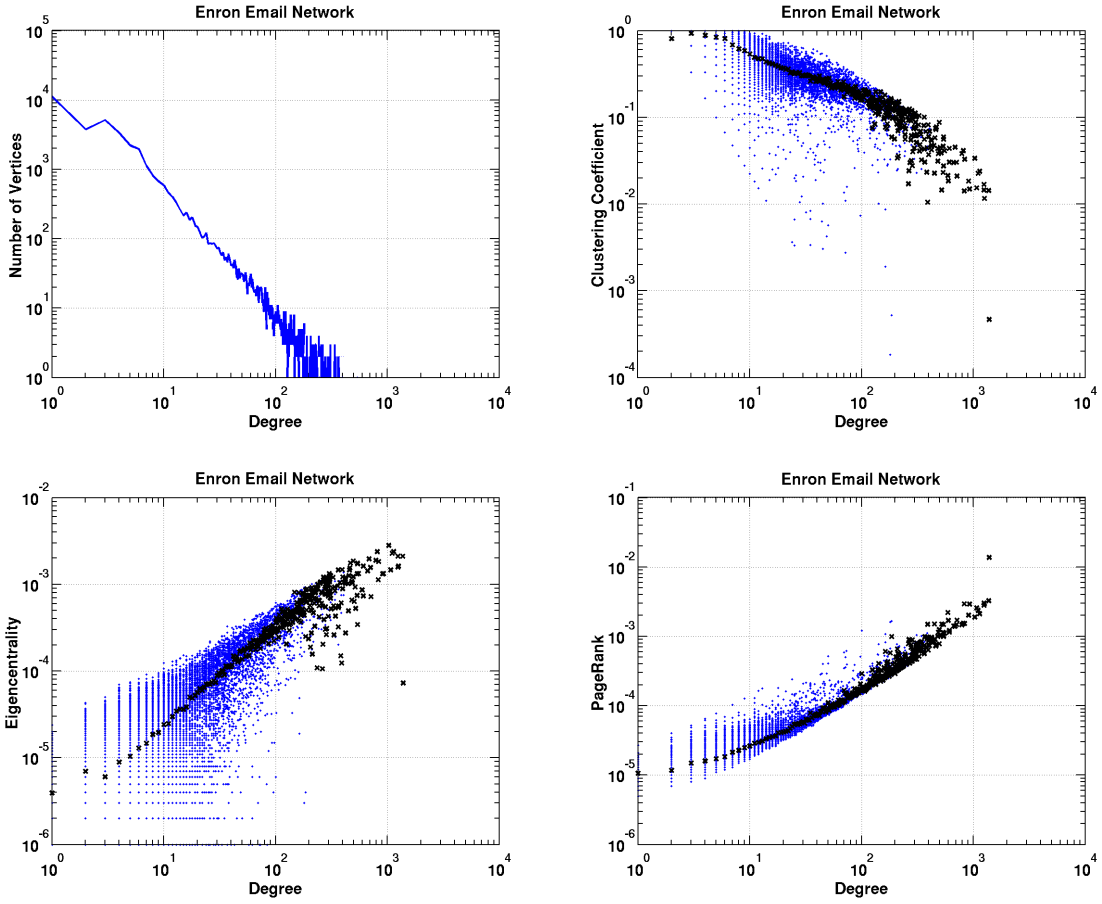

Figure A.6. Vertex statistics of the Enron email network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

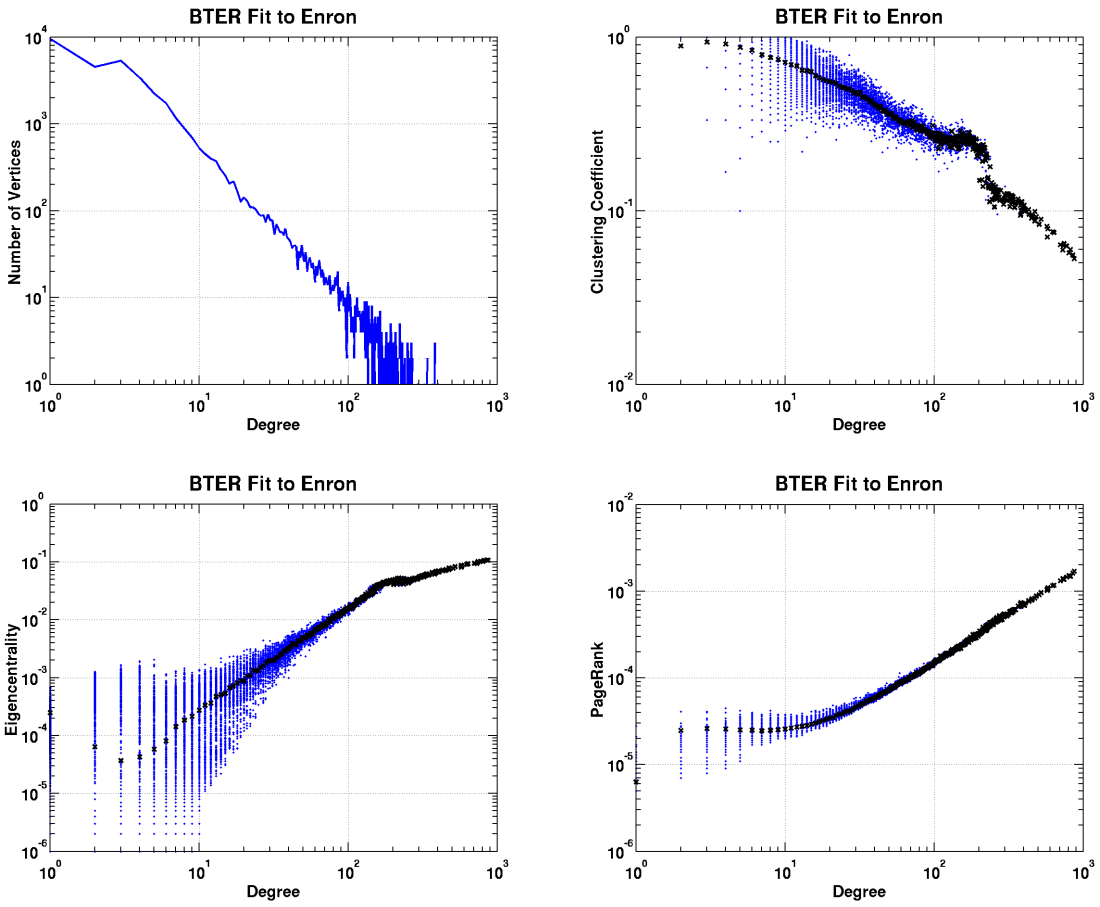

Figure A.7. Vertex statistics of the BTER simulation (fit to Enron email graph) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

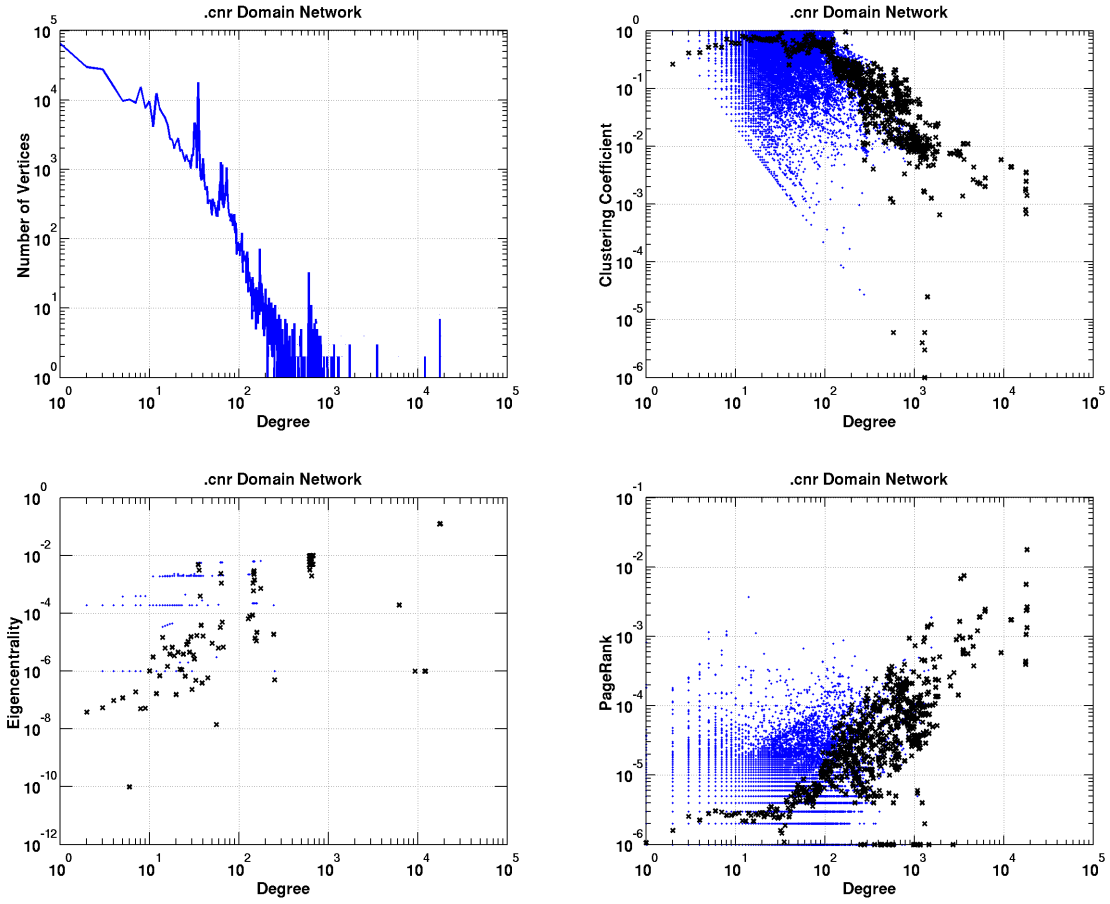

Figure A.8. Vertex statistics of the .cnr domain dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

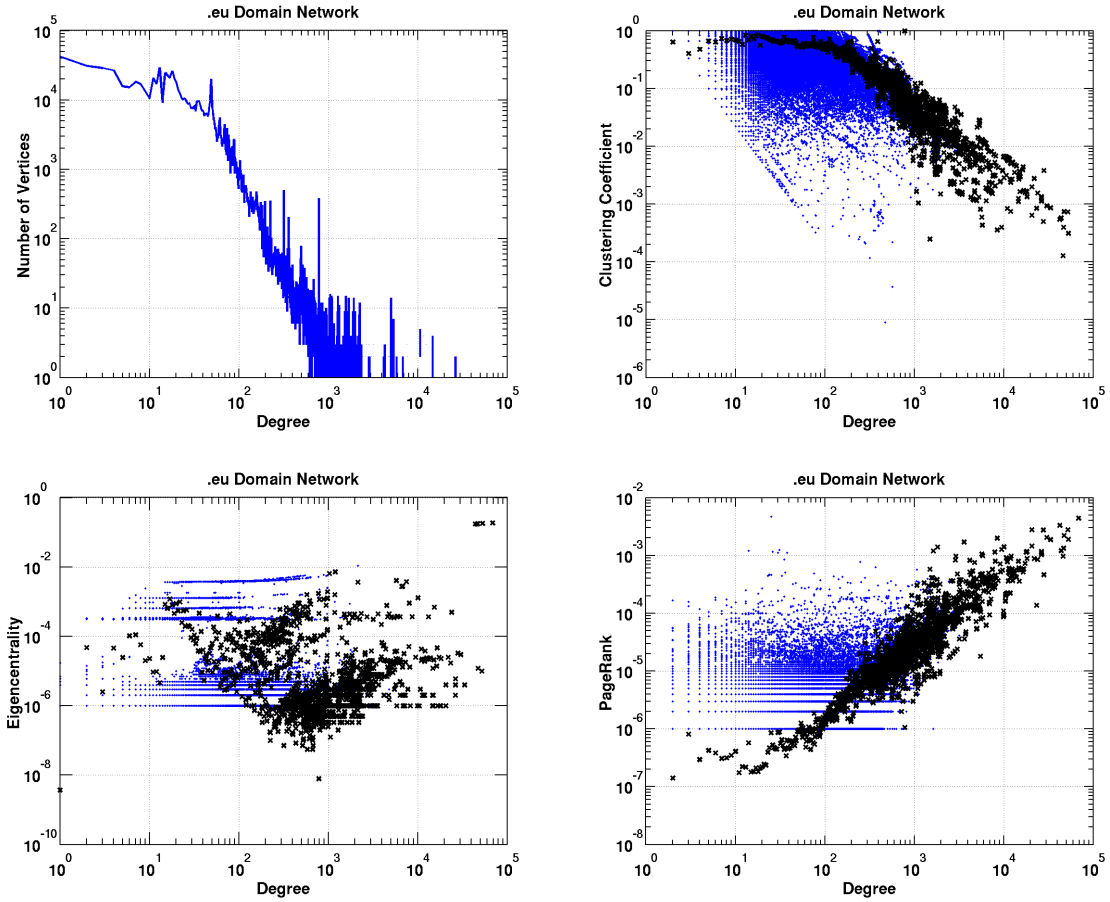

Figure A.9. Vertex statistics of the .eu domain dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

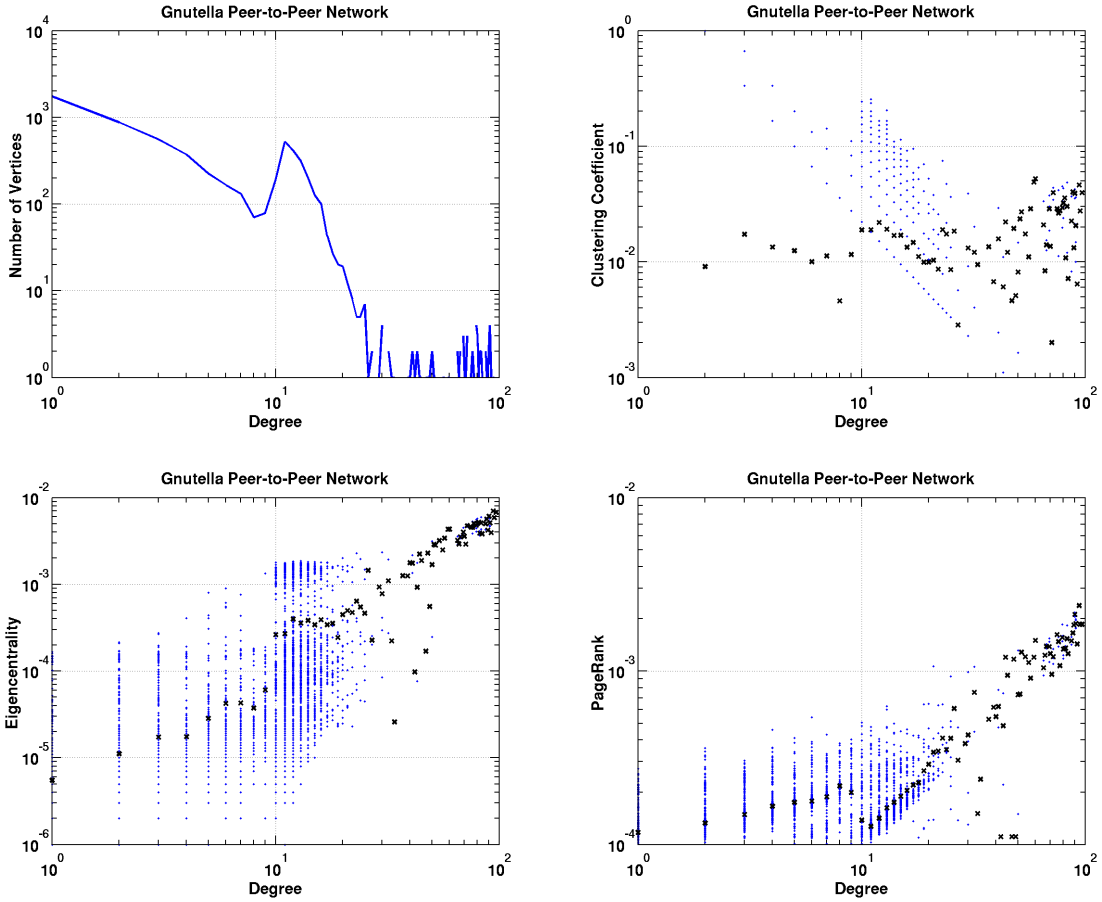

Figure A.10. Vertex statistics of the Gnutella peer-to-peer network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

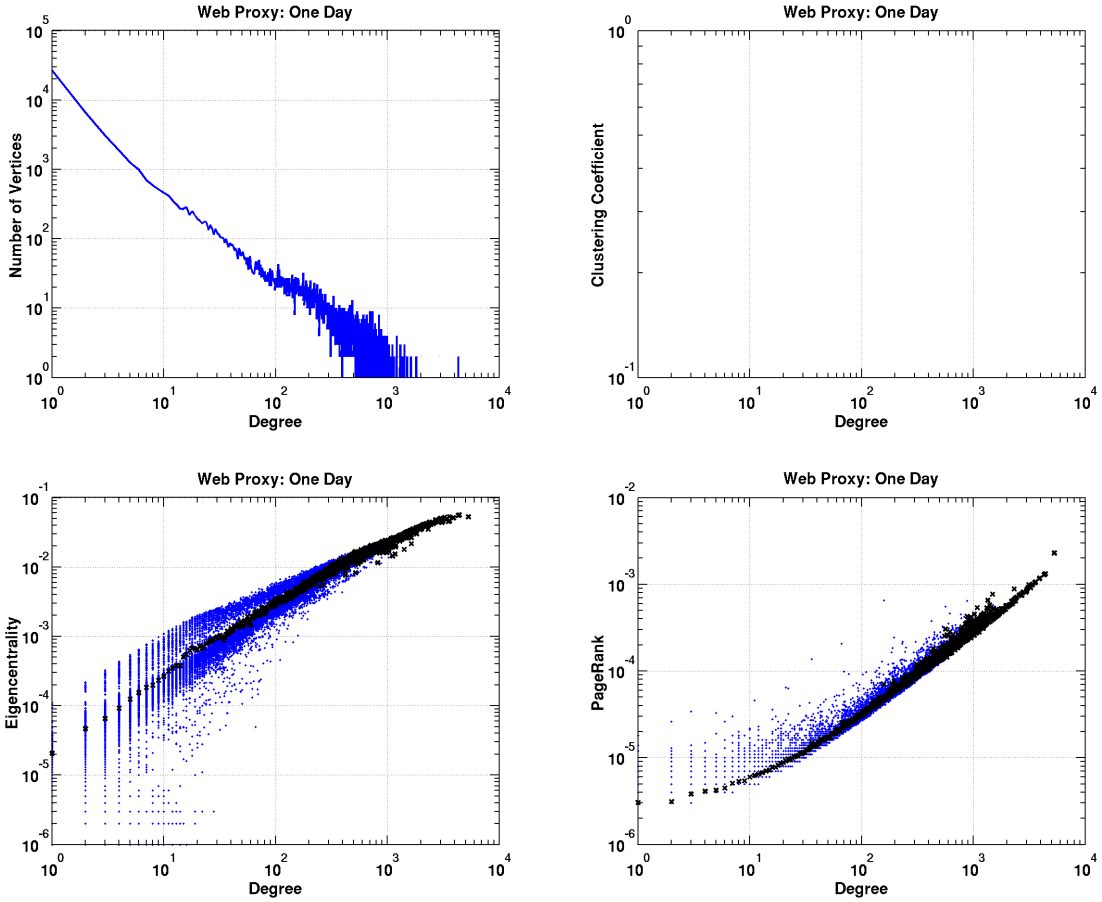

Figure A.11. Vertex statistics of the web proxy (one day) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

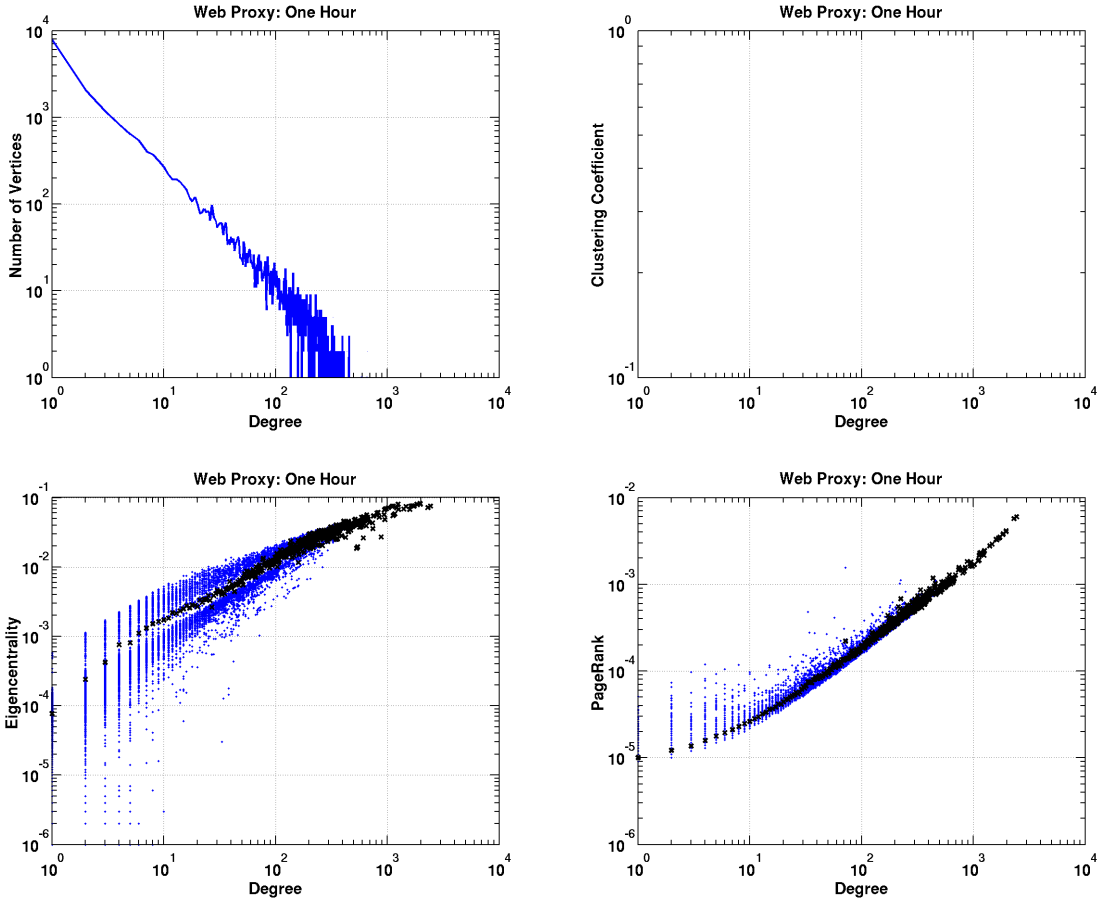

Figure A.12. Vertex statistics of the web proxy (one hour) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

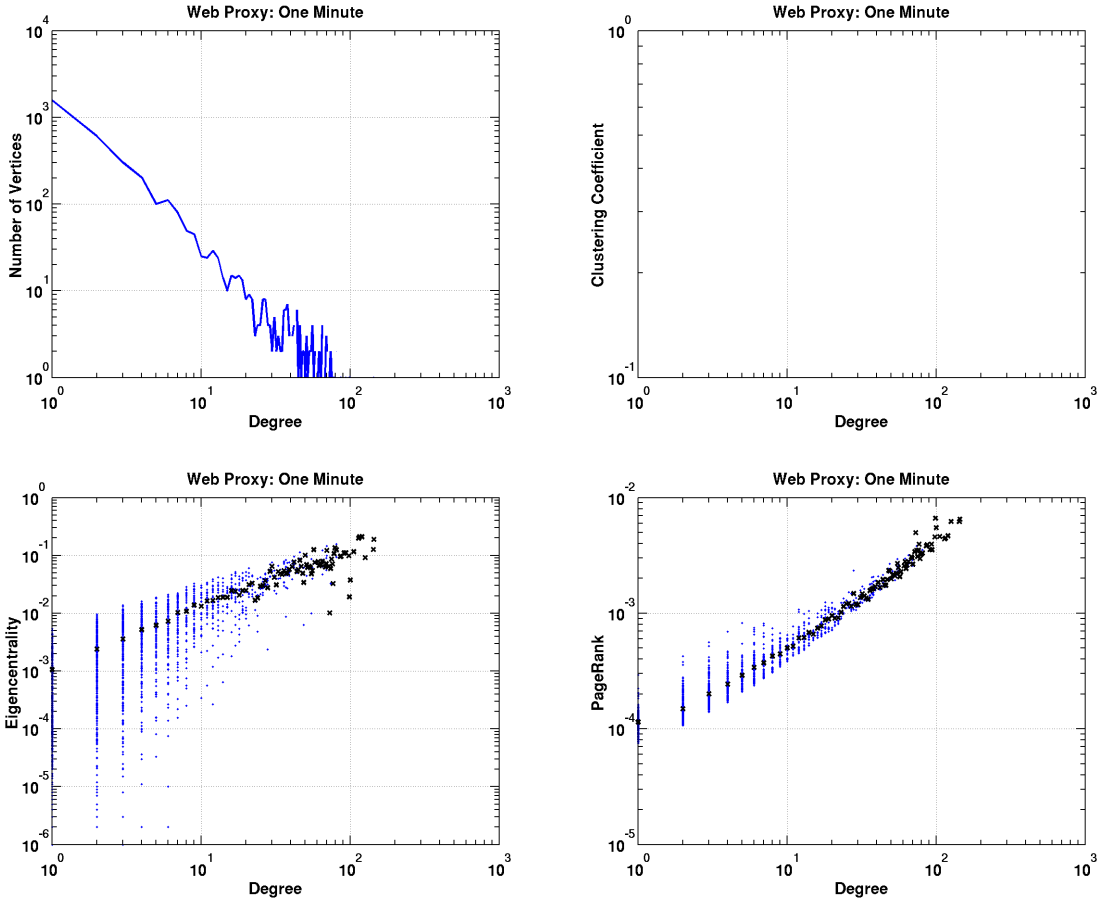

Figure A.13. Vertex statistics of the web proxy (one minute) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

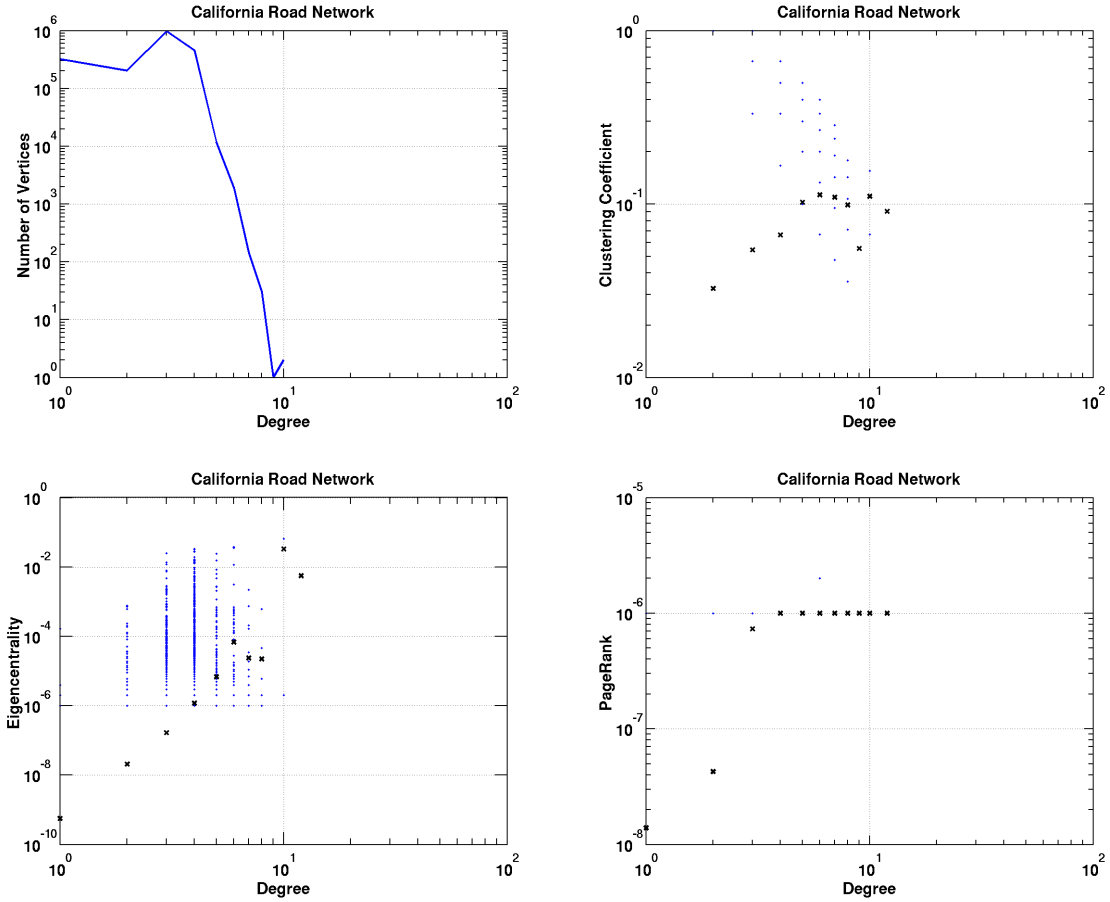

Figure A.14. Vertex statistics of the California road network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

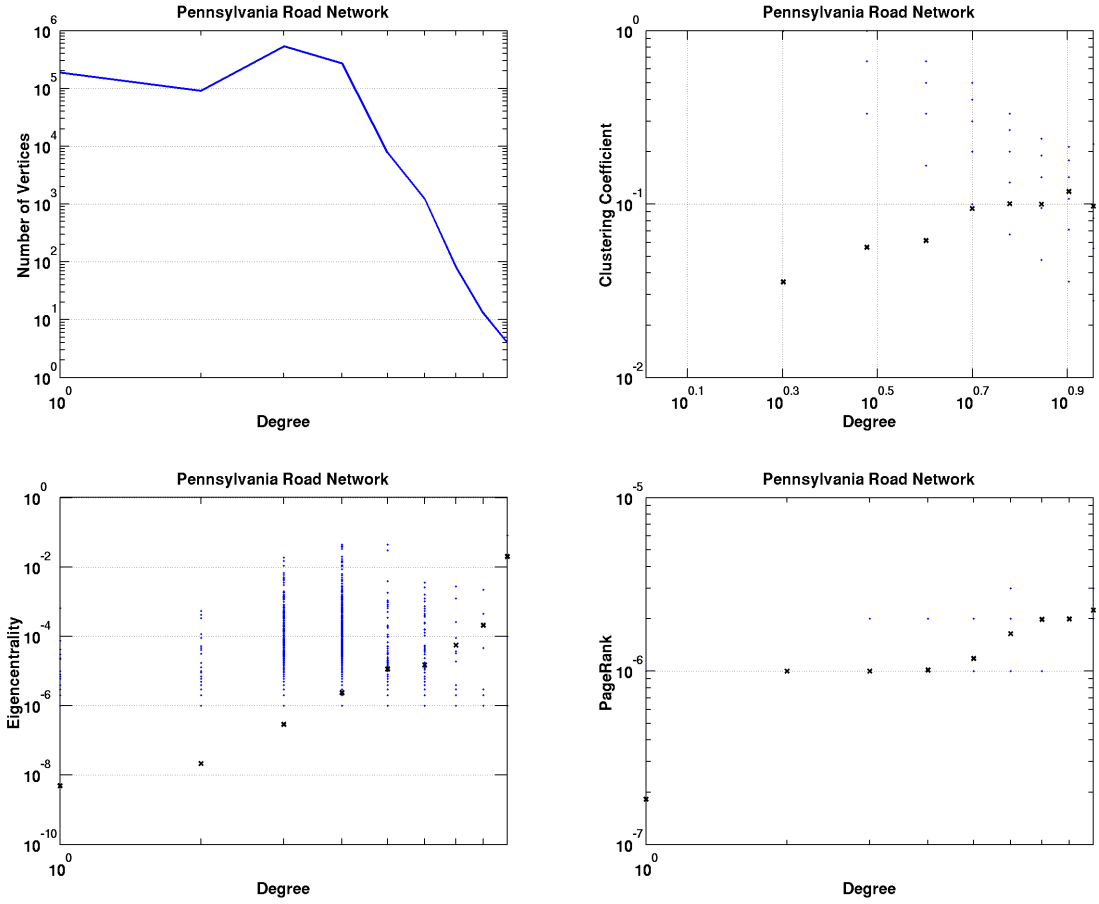

Figure A.15. Vertex statistics of the Pennsylvania road network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

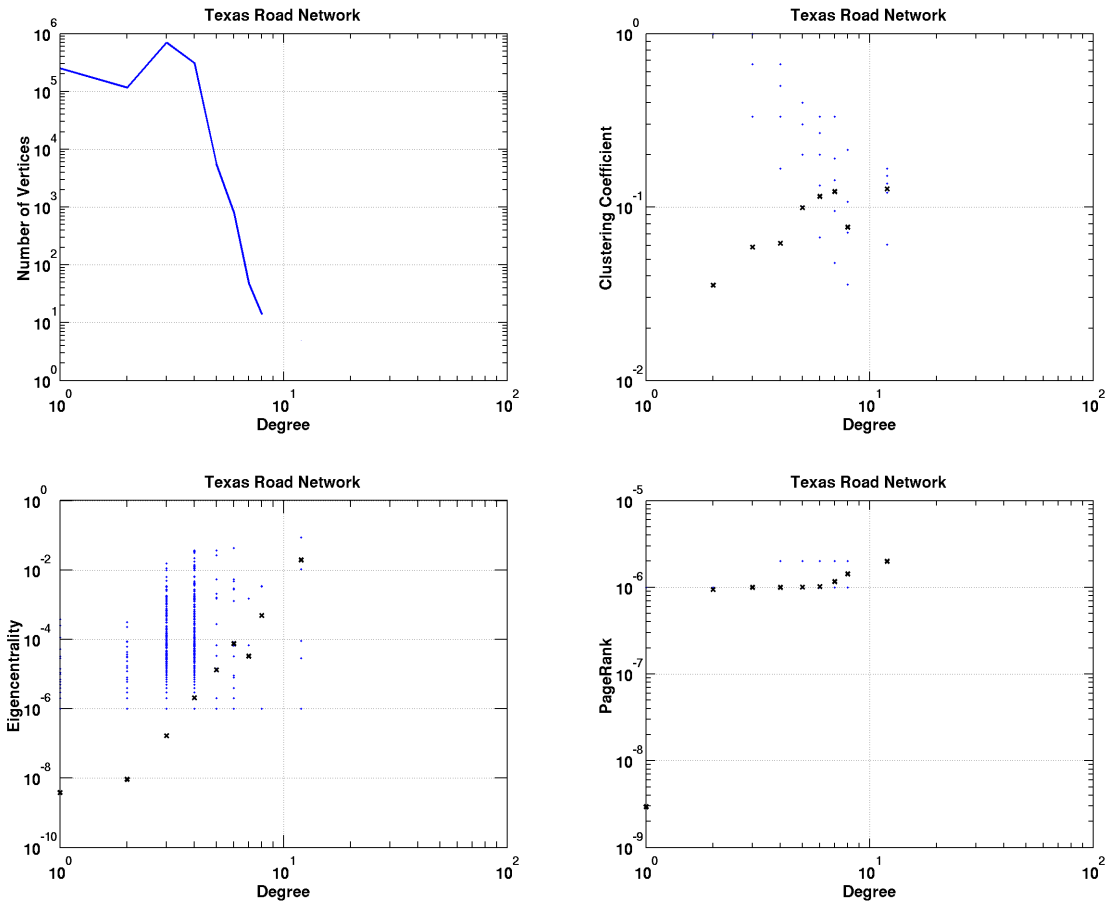

Figure A.16. Vertex statistics of the Texas road network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

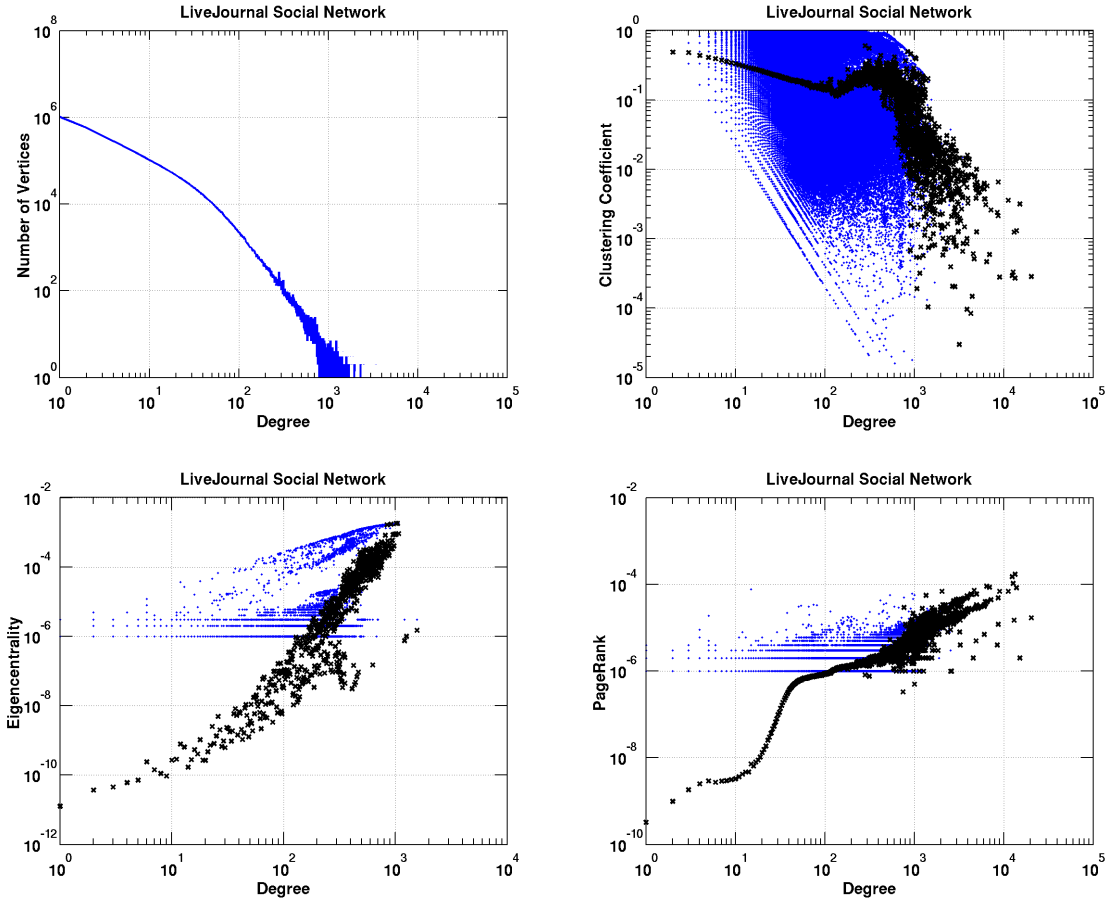

Figure A.17. Vertex statistics of the LiveJournal social network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

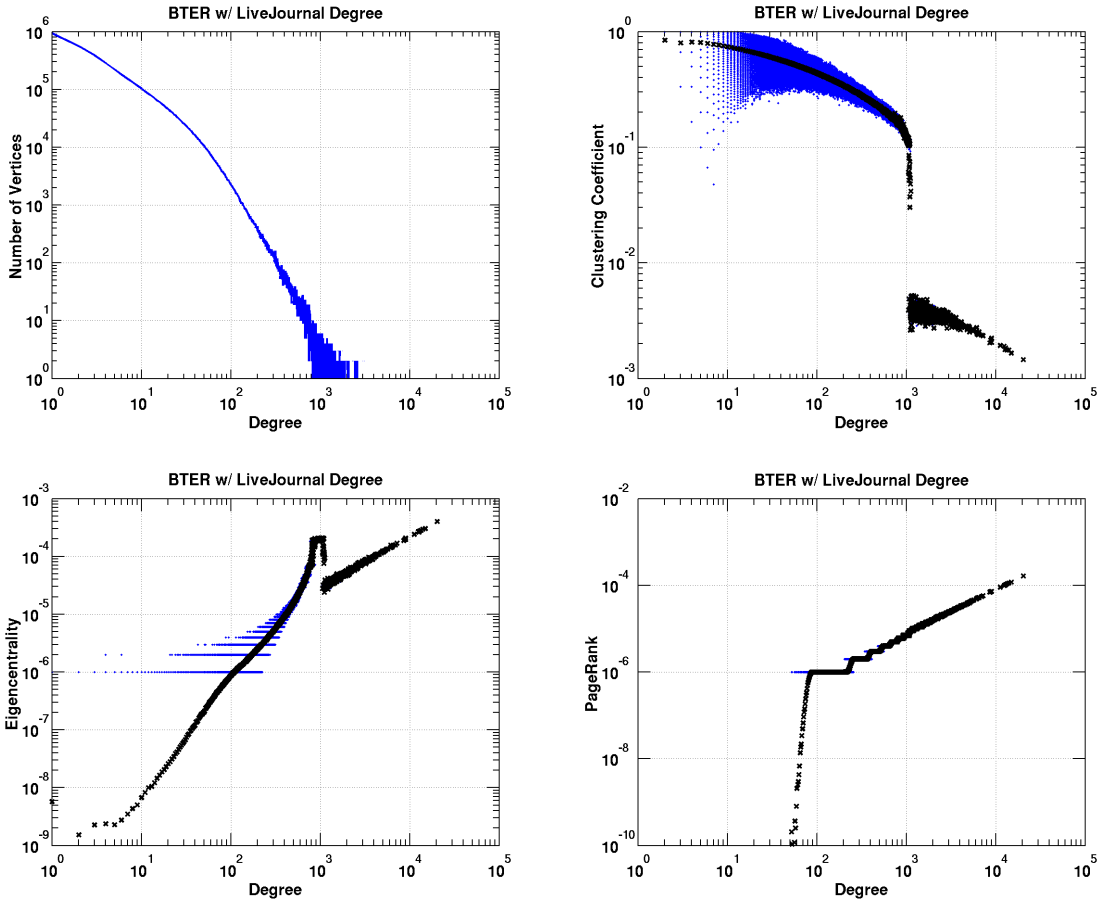

Figure A.18. Vertex statistics of the BTER simulation (with LiveJournal degree distribution) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

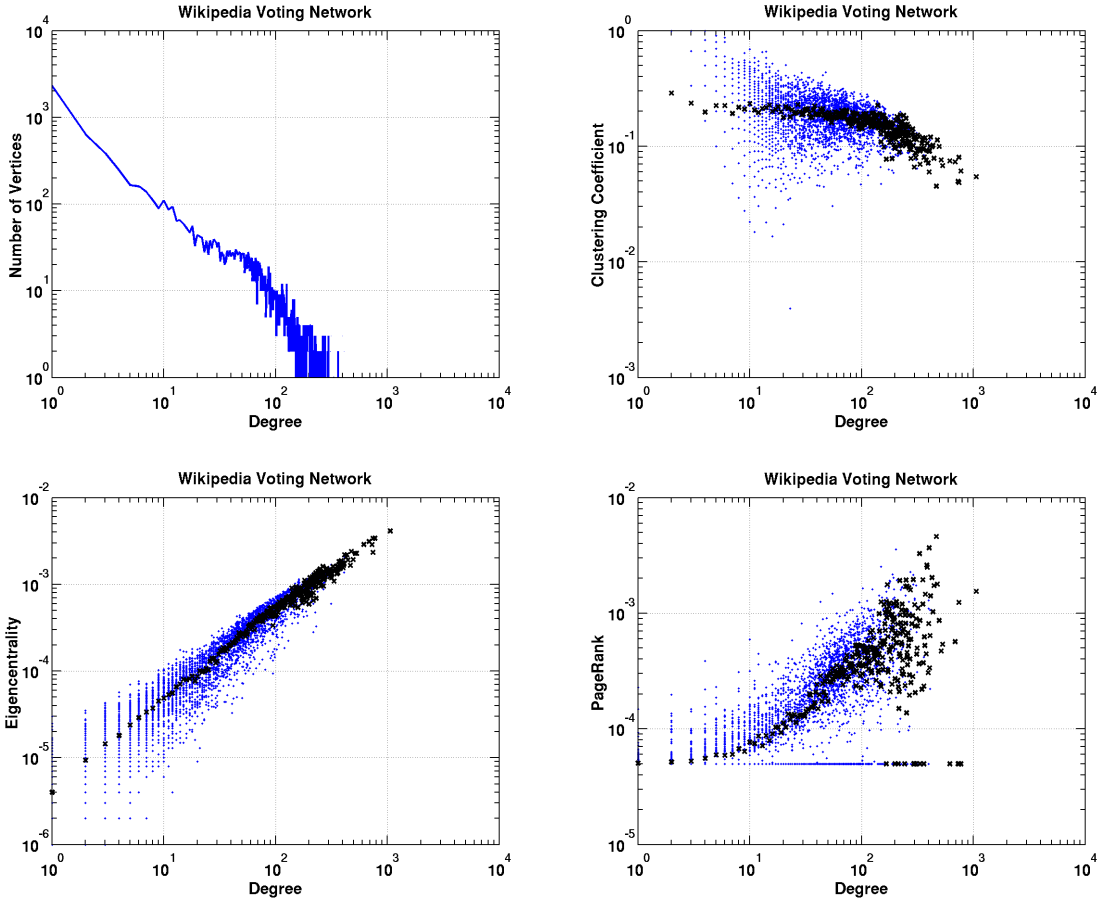

Figure A.19. Vertex statistics of the Wikipedia voting network dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

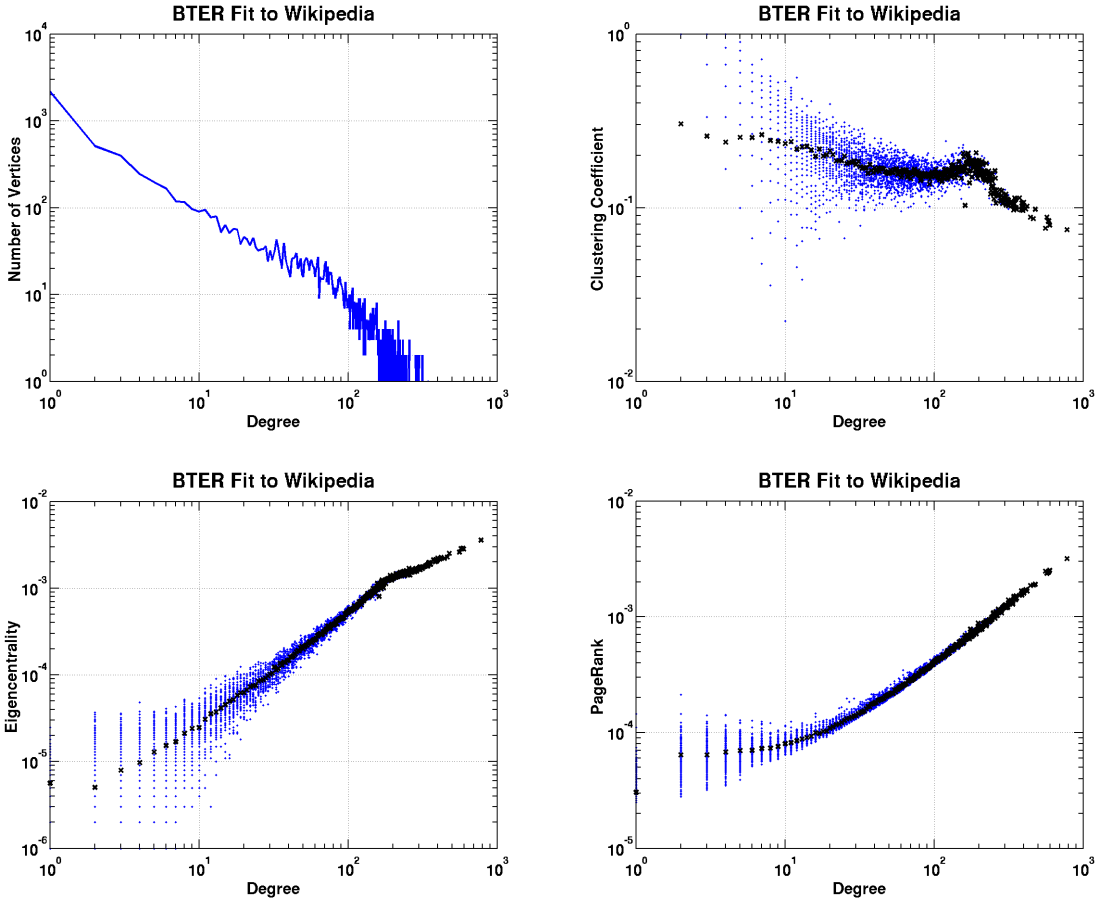

Figure A.20. Vertex statistics of the BTER simulation (fit to Wikipedia voting graph) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

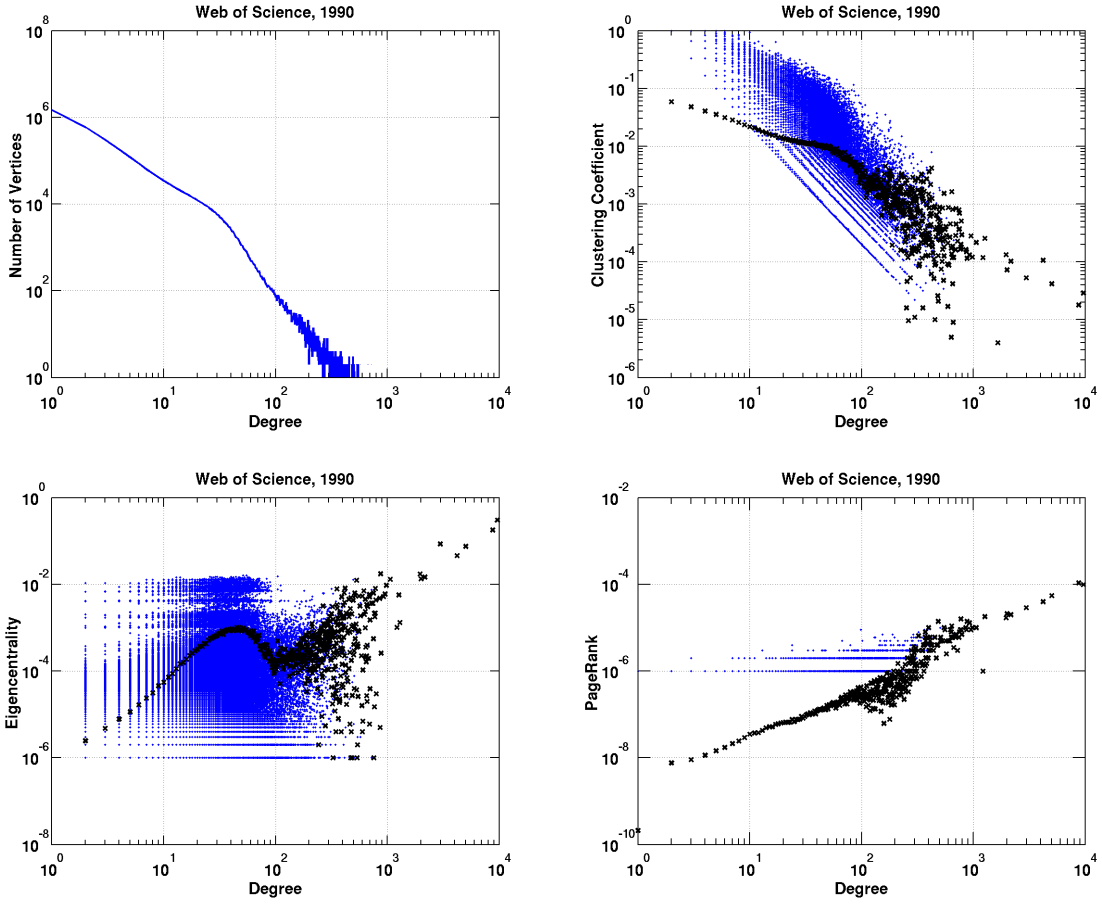

Figure A.21. Vertex statistics of the Web of Science citation network (1990) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

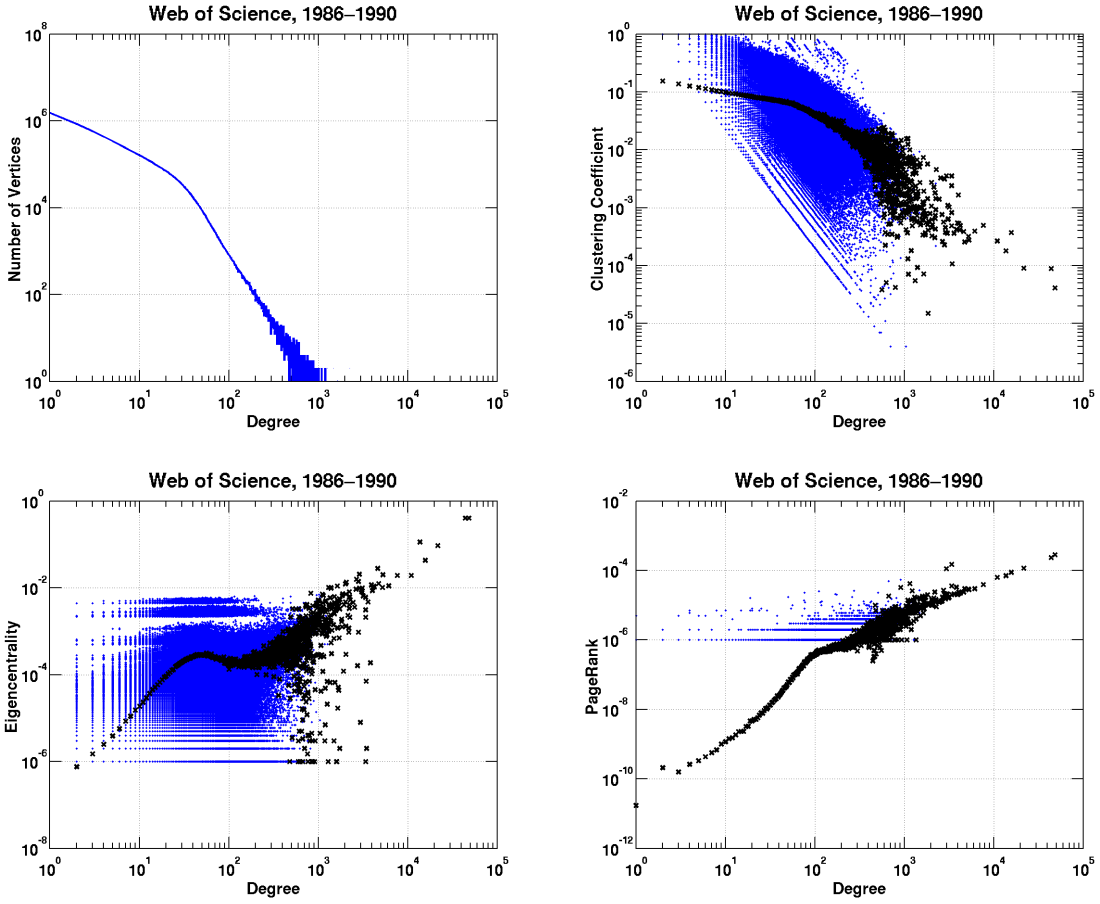

Figure A.22. Vertex statistics of the Web of Science citation network (1986–1990) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

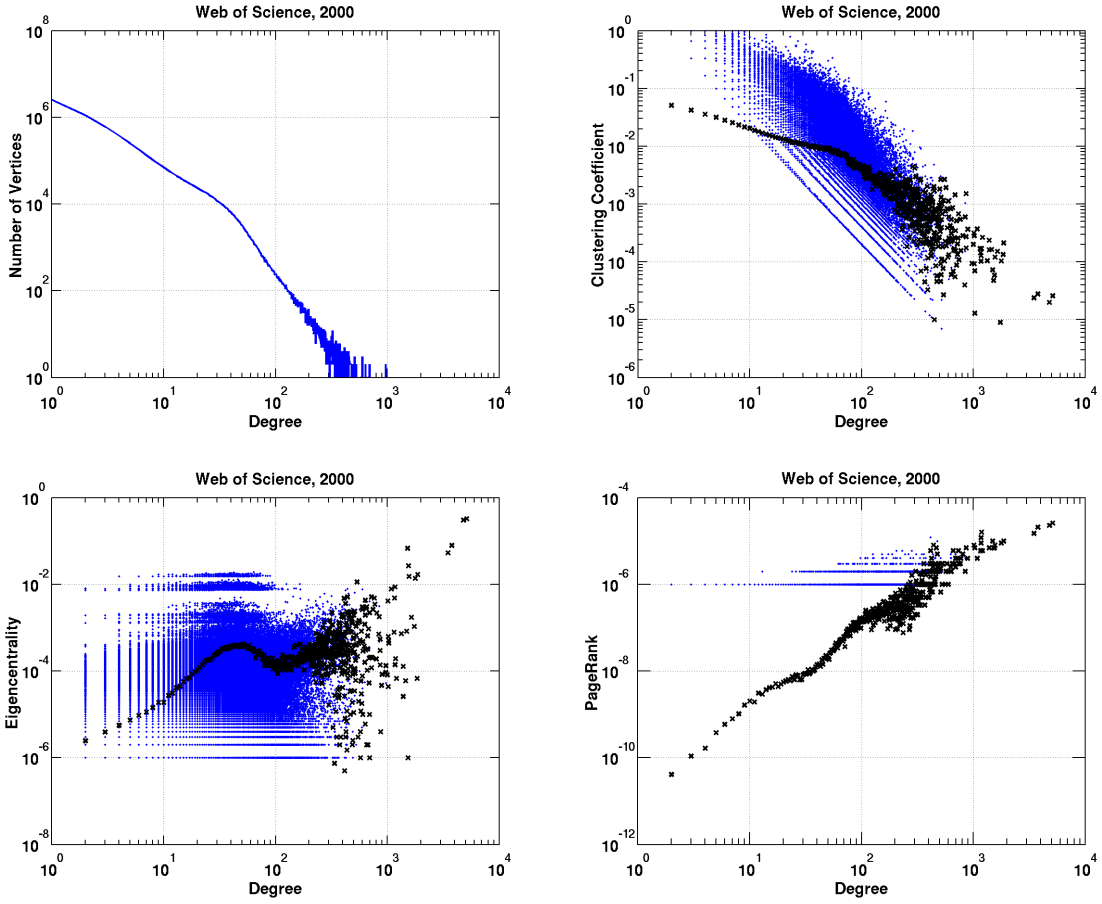

Figure A.23. Vertex statistics of the Web of Science citation network (2000) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

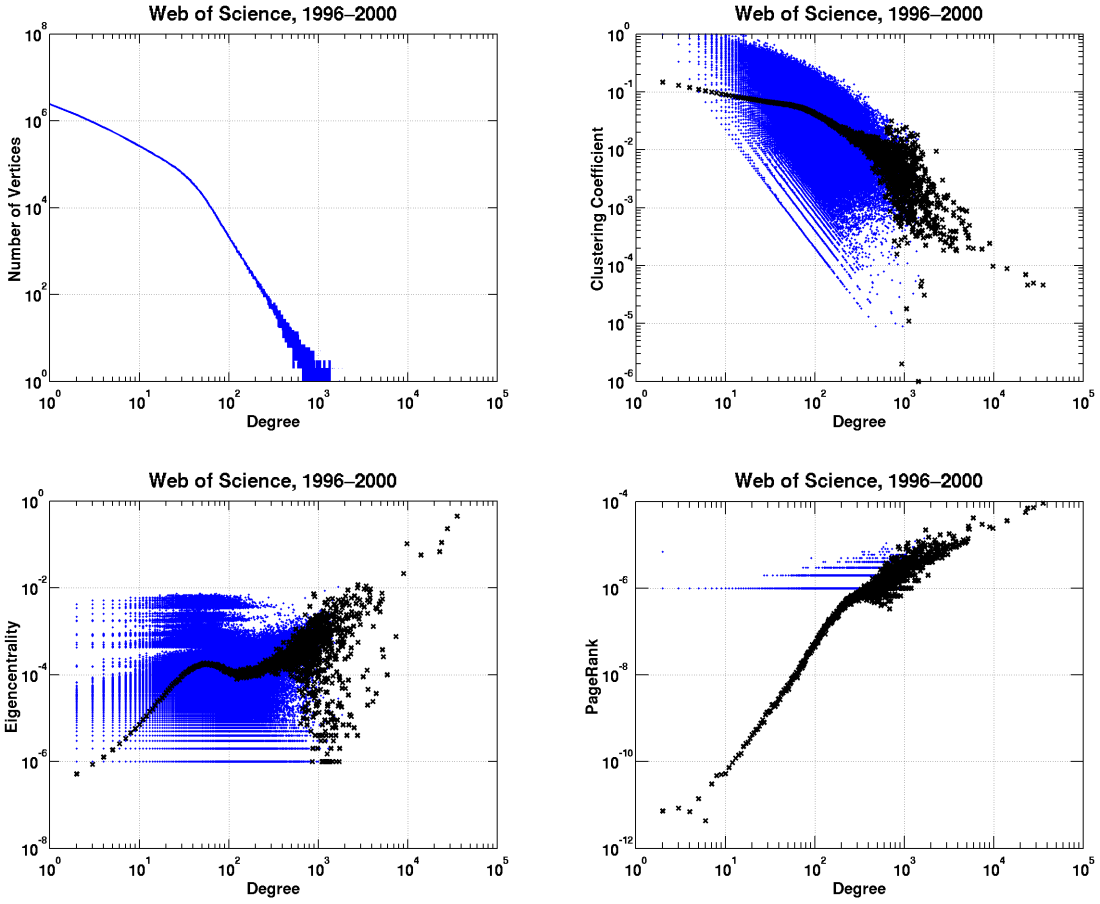

Figure A.24. Vertex statistics of the Web of Science citation network (1996–2000) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

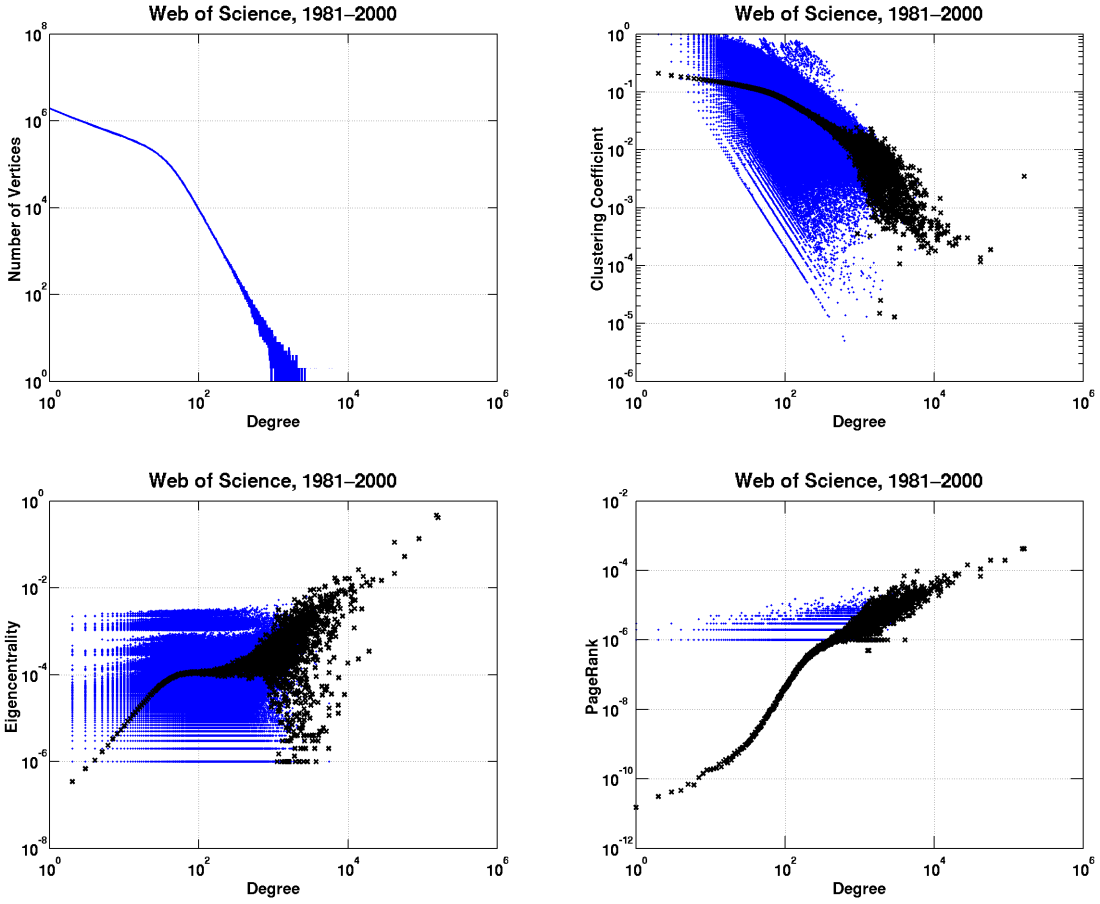

Figure A.25. Vertex statistics of the Web of Science citation network (1981–2000) dataset: degree (upper left), clustering coefficient (upper right), eigencentrality (lower left), and PageRank (lower right).

REFERENCES

[1] B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Toward signal processing theory for graphs and non-Euclidean data," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (2010), pp. 5414–5417.

[2] B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Subgraph detection using eigenvector L1 norms," in J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Inform. Process. Syst. 23*, pp. 1633–1641 (2010).

[3] B.A. Miller, M.S. Beard, and N.T. Bliss, "Matched filtering for subgraph detection in dynamic networks," in *Proc. IEEE Statistical Signal Process. Workshop* (2011), pp. 509–512.

[4] N. Singh, B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Anomalous subgraph detection via sparse principal component analysis," in *Proc. IEEE Statistical Signal Process. Workshop* (2011), pp. 485–488.

[5] B.A. Miller, N.T. Bliss, N. Arcolano, M.S. Beard, J. Kepner, M.C. Schmidt, and E.M. Rutledge, "Very large graphs for information extraction: Summary of first-year proof-of-concept study," MIT Lincoln Laboratory, Project Report VLG-1 (2013).

[6] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proc. KDD* (2006), pp. 631–636.

[7] M.S. Handcock and K.J. Gile, "Modeling social networks from sampled data," *Ann. Appl. Stat.* 4(1), 5–25 (2010).

[8] Y. Zhang, E.D. Kolaczyk, and B.D. Spencer, "Estimating network degree distributions under sampling: An inverse problem, with applications to monitoring social media networks," (2013), preprint: arXiv:1305.4977v1.

[9] S. Bhowmick, S. Srinivasan, and V. Ufimtsev, "Evaluating noise in complex networks," *SIAM Conf. Computational Sci. and Eng.* (2013).

[10] E.D. Kolaczyk, "Quantification of uncertainty in network summary statistics," *SIAM Conf. Computational Sci. and Eng.* (2013).

[11] E. Fields and T.Y. Chen, "On the resilience of graph clusterings," *SIAM Conf. Computational Sci. and Eng.* (2013).

[12] A. Adiga, H. Mortveit, C. Kuhlman, and A. Vullikanti, "Impact of graph perturbations on structural and dynamical properties," *SIAM Conf. Computational Sci. and Eng.* (2013).

[13] B.A. Miller, N. Arcolano, and N.T. Bliss, "Efficient anomaly detection in dynamic, attributed graphs," in *Proc. IEEE Int. Conf. Intelligence and Security Informatics* (2013), pp. 179–184.

[14] C. Seshadhri, T.G. Kolda, and A. Pinar, "Community structure and scale-free collections of Erdős–Rényi graphs," *Phys. Rev. E* 85(5), 056109 (2012).

[15] B.A. Miller, N. Arcolano, M.S. Beard, J. Kepner, M.C. Schmidt, N.T. Bliss, and P.J. Wolfe, "A scalable signal processing architecture for massive graph analysis," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (2012), pp. 5329–5332.

[16] N. Arcolano and B.A. Miller, "Statistical models and methods for anomaly detection in large graphs," *SIAM Ann. Meeting* (2012), minisymposium *Massive Graphs: Big Compute Meets Big Data*.

[17] E.M. Rutledge, B.A. Miller, and M.S. Beard, "Benchmarking parallel eigen decomposition for residuals analysis of very large graphs," in *Proc. IEEE High Performance Extreme Computing Conf.* (2012).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
21 September 2015	Project Report	

4. TITLE AND SUBTITLE

Very Large Graphs for Information Extraction (VLG)
Detection and Interference in the Presence of Uncertainty

5a. CONTRACT NUMBER
FA8721-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Benjamin A. Miller, Nicholas Arcolano, Michelle S. Beard, Leah D. Weiner, Michael Wolf, Albert I. Reuther

5d. PROJECT NUMBER
2088

5e. TASK NUMBER
24

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

VLG-2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intelligence Advanced Research Projects Activity
Office of Incisive Analysis
Washington, DC 20511

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In numerous application domains relevant to the Department of Defense and the Intelligence Community, data of interest take the form of entities and the relationships between them, and these data are commonly represented as graphs. Under the Very Large Graphs for Information Extraction effort—a one-year proof-of-concept study—MIT LL developed novel techniques for anomalous subgraph detection, building on tools in the signal processing research literature. This report documents the technical results of this effort. Two datasets—a snapshot of Thompson Reuters' Web of Science database and a stream of web proxy logs—were parsed, and graphs were constructed from the raw data. From the phenomena in these datasets, several algorithms were developed to model the dynamic graph behavior, including a preferential attachment mechanism with memory, a streaming filter to model a graph as a weighted average of its past connections, and a generalized linear model for graphs where connection probabilities are determined by additional side information or metadata. A set of metrics was also constructed to facilitate comparison of techniques. The study culminated in a demonstration of the algorithms on the datasets of interest, in addition to simulated data. Performance in terms of detection, estimation, and computational burden was measured according to the metrics. Among the highlights of this demonstration were the detection of emerging coauthor clusters in the Web of Science data, detection of botnet activity in the web proxy data after 15 minutes (which took 10 days to detect using state-of-the-practice techniques), and demonstration of the core algorithm on a simulated 1-billion-vertex graph using a commodity computing cluster.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	Same as report	62	

This page intentionally left blank.